Size-Constrained Tree Decompositions

Bi Li, Fatima Zahra Moataz, Nicolas Nisse, Karol Suchan

Size-Constrained Tree Decompositions ⋆

Bi Li[1,2,3], Fatima Zahra Moataz[2,1], Nicolas Nisse[1,2], and Karol Suchan[4,5]

[1] Inria, France
[2] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, Sophia Antipolis, France
[3] Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China
[4] FIC, Universidad Adolfo Ibáñez, Santiago, Chile
[5] WMS, AGH, University of Science and Technology, Krakow, Poland

Abstract. *Tree-Decompositions* are the corner-stone of many dynamic programming algorithms for solving graph problems. Since the complexity of such algorithms generally depends exponentially on the *width* (size of the *bags*) of the decomposition, much work has been devoted to compute tree-decompositions with small width. However, practical algorithms computing tree-decompositions only exist for graphs with *treewidth* less than 4. In such graphs, the time-complexity of dynamic programming algorithms based on tree-decompositions is dominated by the *size* (number of bags) of the tree-decompositions. It is then interesting to minimize the size of the tree-decompositions. In this report, we consider the problem of computing a tree-decomposition of a graph with width at most k and minimum size. More precisely, we focus on the following problem: given a fixed $k \geq 1$, what is the complexity of computing a tree-decomposition of width at most k with minimum size in the class of graphs with treewidth at most k? We prove that the problem is NP-complete for any fixed $k \geq 4$ and polynomial for $k \leq 2$; for $k = 3$, we show that it is polynomial in the class of trees and 2-connected outerplanar graphs.

1 Introduction

A *tree-decomposition* of a graph [13] is a way to represent G by a family of subsets of its vertex-set organized in a tree-like manner and satisfying some connectivity property. The *treewidth* of G measures the proximity of G to a tree. More formally, a tree decomposition of $G = (V, E)$ is a pair $(T, \mathcal{X})$ where $\mathcal{X} = \{X_t | t \in V(T)\}$ is a family of subsets, called *bags*, of V, and T is a tree, such that:

- $\bigcup_{t \in V(T)} X_t = V$;
- for any edge $uv \in E$, there is a bag X_t (for some node $t \in V(T)$) containing both u and v;
- for any vertex $v \in V$, the set $\{t \in V(T) | v \in X_t\}$ induces a subtree of T.

The *width* of a tree-decomposition $(T, \mathcal{X})$ is $max_{t \in V(T)} |X_t| - 1$ and its *size* is order $|V(T)|$ of T. The treewidth of G, denoted by $tw(G)$, is the minimum width over all possible tree-decompositions of G.

If T is constrained to be a path, $(T, \mathcal{X})$ is called a *path-decomposition* of G. The pathwidth of G, denoted by $pw(G)$, is the minimum width over all possible path-decompositions of G.

Tree-Decompositions are the corner-stone of many dynamic programming algorithms for solving graph problems. As an example, the famous Courcelle's Theorem states that any problem expressible in MSOL can be solved in linear-time in the class of bounded treewidth graphs [6]. Another framework based on graph decompositions is the *bi-dimensionality theory* that allowed the design of sub-exponential-time algorithms for many problems in the class of graphs excluding some fixed graph as a minor (e.g., [7]). Given a tree-decomposition with width w and size n, the time-complexity of most of such dynamic programming algorithms can be expressed as $O(2^w n)$ (or $O(2^{w \log w} n)$ in the case of *global* problems). Therefore, the problem of computing tree-decompositions with small width has drawn much attention in the last decades. It has been extensively studied and investigated from different angles: parametrized complexity, exact or approximation algorithms.

The above mentioned algorithms have mainly a theoretical interest because, on the one hand, their time-complexity exponential depends on the treewidth of graphs and, on the other hand, as far as we know, no practical algorithm exists that computes a "good" tree-decomposition for graphs with treewidth at least 5. However, in case of small (≤ 4)

⋆ This work has been partially supported by European Project FP7 EULER, ANR project Stint (ANR-13-BS02-0007), the associated Inria team AlDyNet, the project ECOS-Sud Chile and a grant from the "Conseil régional Provence Alpes-Côte d'Azur".

treewidth graphs, efficient (i.e., practical) algorithms exist to compute tree-decompositions with optimal width. Moreover, in such case, the time-complexity of above-mentioned dynamic programming algorithms becomes dominated by the size of the tree-decompositions and, therefore, it becomes interesting to minimize it.

In report, we study the problem of computing tree-decompositions with minimum size. Obviously, if the width is not constrained, then the problem is trivial since there always exists a tree-decomposition of a graph with one bag (the full vertex-set). Hence, given a graph G and an integer $k \geq tw(G)$, we consider the problem of minimizing the size of a tree-decomposition of G with width at most k.

Related Work. The problem of computing "good" tree-decompositions has been extensively studied. Computing optimal tree-decomposition - i.e., with width $tw(G)$ - is NP-complete in the class of general graphs G [1]. For any fixed $k \geq 1$, Bodlaender designed an algorithm that computes, in time $O(k^{k^3} n)$, a tree-decomposition of width k of any n-node graph with treewidth at most k [3]. Very recently, a single-exponential (in k) algorithm has been proposed that computes a tree-decomposition with width at most $5k$ in the class of graphs with treewidth at most k [4]. As far as we know, the only practical algorithms for computing optimal tree-decompositions hold for graphs with treewidth at most 1 (trivial since $tw(G) = 1$ if and only if G is a tree), 2 (graphs excluding K_4 as a minor) [16], 3 [2, 11, 12] and 4 [14].

We are not aware of any work dealing with the computation of tree-decompositions with minimum size. In [8], Dereniowski *et al.* consider the problem of size-constrained path-decompositions. Given any positive integer k and any graph G with pathwidth at most k. Let $l_k(G)$ denote the smallest size (length) of a path-decomposition of G with width at most k. For any fixed $k \geq 4$, computing l_k is NP-complete in the class of general graphs and it is NP-complete, for any fixed $k \geq 5$, in the class of connected graphs [8]. Moreover, computing l_k can be solved in polynomial-time in the class of graphs with pathwidth at most k for any $k \leq 3$. Finally, the "dual" problem is also hard: for any fixed $s \geq 2$, it is NP-complete in general graphs to compute the minimum width of a tree-decomposition with size s [8][6].

Our results. Let k be any positive integer and G be any graph. If $tw(G) > k$, let us set $s_k(G) = \infty$. Otherwise, let $s_k(G)$ denote the minimum size of a tree-decomposition of G with width at most k. See a simple example in Fig. 1. We first prove in Section 2 that, for any (fixed) $k \geq 4$, the problem of computing s_k is NP-hard in the class of graphs with treewidth at most k. Moreover, the computation of s_k for $k \geq 5$ is NP-hard in the class of connected graphs with treewidth at most k. Furthermore, the computation of s_4 is NP-complete in the class of planar graphs with treewidth 3. In Section 3, we present a general approach for computing s_k for any $k \geq 1$. In the rest of the report, we prove that computing s_2 can be solved in polynomial-time. Finally, we prove that s_3 can be computed in polynomial time in the class of trees and 2-connected outerplanar graphs.

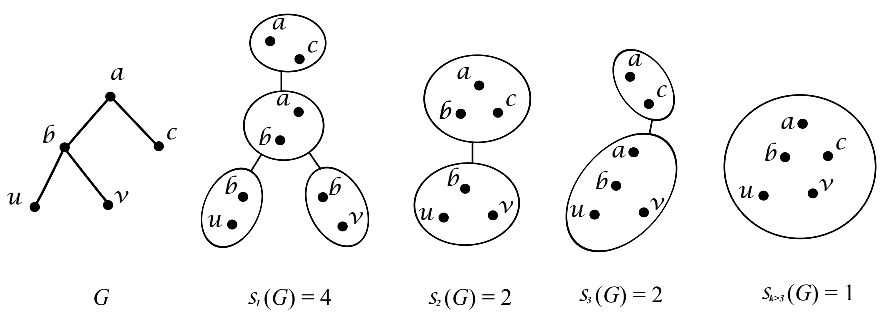


Fig. 1. Given a tree G with five vertices, for any $k \geq 1$, a minimum size tree decomposition of width at most k is shown. So we see that $s_1(G) = 4$, $s_2(G) = s_3(G) = 2$, $s_{k>3}(G) = 1$.

2 NP-hardness in the class of bounded treewidth graphs

In this section, we prove that:

[6] This result was proved in [8] in terms of path-decomposition but it is straightforward to extend it to tree-decomposition.

Theorem 1. *For any fixed integer $k \geq 4$ (resp., $k \geq 5$), the problem of computing s_k is NP-complete in the class of graphs (resp., of connected graphs) with treewidth at most k.*

Note that the corresponding decision problem is clearly in NP. Hence, we only need to prove it is NP-hard.

Our proof mainly follows the one of [8] for size-constrained path-decompositions. Hence, we recall here the two steps of the proof in [8]. First, it is proved that, if computing l_k is NP-hard for any $k \geq 1$ in general graphs, then the computation of l_{k+1} is NP-hard in the class of connected graphs. Second, it is shown that computing l_4 is NP-hard in general graphs with pathwidth 4. In particular, this implies that computing l_5 is NP-hard in the class of connected graphs with pathwidth 5. The second step consists of a reduction from the 3-PARTITION problem [9] to the one of computing l_4. Precisely, for any instance $\mathcal{I}$ of 3-PARTITION, a graph $G_{\mathcal{I}}$ is built such that $\mathcal{I}$ is a YES instance if and only if $l_4(G_{\mathcal{I}})$ equals a defined value $\ell_{\mathcal{I}}$.

Our contribution consists first in showing that the first step of [8] directly extends to the case of tree-decompositions. That is, it directly implies that, if computing s_k is NP-hard for some $k \geq 4$ in general graphs, then so is the computation of s_{k+1} in the class of connected graphs. Our main contribution of this section is to show that, for the graphs $G_{\mathcal{I}}$ built in the reduction proposed in [8], any tree-decomposition of $G_{\mathcal{I}}$ with width at most 4 and minimum size is a path decomposition. Hence, in this class of graphs, $l_4 = s_4$ and, for any instance $\mathcal{I}$ of 3-PARTITION, $\mathcal{I}$ is a YES instance if and only if $s_4(G_{\mathcal{I}})$ equals a defined value $\ell_{\mathcal{I}}$. We describe the details in what follows.

Lemma 1. *If the problem of computing s_k for an integer $k \geq 1$ is NP-complete in general graphs, then the computation of s_{k+1} is NP-complete in the class of connected graphs.*

Proof. Let G be any graph. We construct an auxiliary connected graph G' from G by adding a vertex a adjacent to all vertices in $V(G)$. Given two integers $k, s \geq 1$, in the following, we prove that there is a tree decomposition of G with width at most k and size at most s if and only if there is a tree decomposition of G' with width at most $k+1$ and size at most s.

First, assume that $(T, \mathcal{X})$ is a tree decomposition of G with width at most k and size at most s. Add a in each bag of $\mathcal{X}$. Then we obtain a tree decomposition of G' with width at most $k+1$ and size at most s.

Now let $(T', \mathcal{X}')$ be a tree decomposition of G' with width at most $k+1$ and size at most s. We are going to find a tree decomposition of G with width at most k and size at most s. Let $\mathcal{X}_a$ be the set of all bags in $\mathcal{X}'$ containing a. Let T_a be the subtree of T' induced by the bags in $\mathcal{X}_a$. Every vertex $v \in V(G)$ is contained in a bag in $\mathcal{X}_a$ because $va \in E(G')$. For any edge $uv \in E(G)$, there is a bag $X \supseteq \{a, u, v\}$ in $\mathcal{X}'$ since $\{a, u, v\}$ induces a clique in G'. So $X \in \mathcal{X}_a$. Delete a in each bag of $\mathcal{X}_a$ and denote by $\mathcal{X}^-$ the obtained set of bags. So $(T_a, \mathcal{X}^-)$ is a tree decomposition of G with width at most k and size at most s.

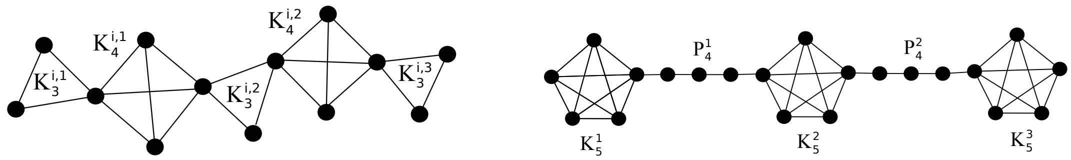
Before doing the reduction from the 3-PARTITION problem to the problem of computing s_4, let us first recall its definition.

Definition 1. [3-PARTITION]
Instance: A multiset S of $3m$ positive integers $S = (w_1, \dots, w_{3m})$ and an integer b.
Question: Is there a partition of the set $\{1, \dots, 3m\}$ into m sets $S_1, \dots, S_m$ such that $\sum_{i \in S_j} w_i = b$ for each $j = 1, \dots, m$?

This problem is NP-complete even if $|S_j| = 3$ for all $j = 1, \dots, m$ [9].

Given an instance of 3-PARTITION, in the following, we construct a disconnected graph $G(S, b)$ as in [8]. First, for each $i \in \{1, \dots, 3m\}$, we construct a connected graph H_i as follows. Take w_i copies of K_3, denoted by $K_3^{i,q}$, $q = 1, \dots, w_i$, and $w_i - 1$ copies of K_4, denoted by $K_4^{i,q}$, $q = 1, \dots, w_i - 1$ (the copies are mutually disjoint). Then for each $q = 1, \dots, w_i - 1$, we identify two different vertices of $K_4^{i,q}$ with a vertex of $K_3^{i,q}$ and with a vertex of $K_3^{i,q+1}$, respectively. This is done in such a way that each vertex of each $K_3^{i,q}$ is identified with at most one vertex from other cliques. Informally the cliques form a 'chain' in which the cliques of size 3 and 4 alternate. See Figure 2(a) for an example of H_i where $w_i = 3$.

Second, we construct a graph $H_{m,b}$ as follows. Take $m+1$ copies of K_5, denoted by $K_5^1, \dots, K_5^{m+1}$, and m copies of the path graph P_b of length b (P_b has b edges and $b+1$ vertices), denoted by $P_b^1, \dots, P_b^m$. (Again, the copies are taken to be mutually disjoint.) Now, for each $j = 1, \dots, m$, identify one of the endpoints of P_b^j with a vertex of



(a) H_i, where $w_i = 3$.

(b) $H_{2,4}$

Fig. 2. Examples of gadgets in graph $G(S, b)$.

K_5^j, and identify the other endpoint with a vertex of K_5^{j+1}. Moreover, do this in a way that ensures that, for each j, no vertex of K_5^j is identified with the endpoints of two different paths. See Figure 2(b) for an example of $H_{2,4}$.

Let $G(S, b)$ be the graph obtained by taking the disjoint union of the graphs $H_1, \ldots, H_{3m}$ and the graph $H_{m,b}$. In the following, we prove that there is a tree decomposition of $G(S, b)$ of width 4 and size at most $s = 1 - 2m + 2\sum_{i=1}^{3m} w_i$ if and only if there is a partition of the set $\{1, \ldots, 3m\}$ into m sets $S_1, \ldots, S_m$ such that $\sum_{i \in S_j} w_i = b$ for each $j = 1, \ldots, m$ in the instance of 3-PARTITION.

In Lemma 2.2 of [8], a path decomposition of $G(S, b)$ of width 4 and length $1 - 2m + 2\sum_{i=1}^{3m} w_i$ is constructed if there is a partition of the set $\{1, \ldots, 3m\}$ into m sets $S_1, \ldots, S_m$ such that $\sum_{i \in S_j} w_i = b$ for each $j = 1, \ldots, m$ in the instance of 3-PARTITION. Obviously, this path decomposition is also a tree decomposition of $G(S, b)$ of width 4 and size s. So we have the following lemma.

Lemma 2. *Given a multiset S of $3m$ positive integers $S = (w_1, \ldots, w_{3m})$ and an integer b, if there is a partition of the set $\{1, \ldots, 3m\}$ into m sets $S_1, \ldots, S_m$ such that $\sum_{i \in S_j} w_i = b$ for each $j = 1, \ldots, m$, then $G(S, b)$ has a tree decomposition of width at most 4 and size at most $s = 1 - 2m + 2\sum_{i=1}^{3m} w_i$.*

Now we prove the other direction.

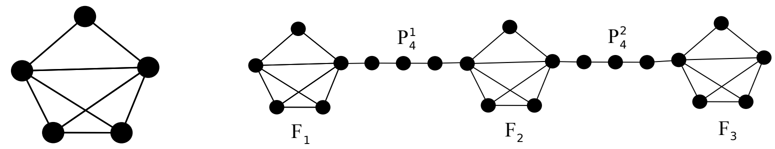
Lemma 3. *If $G(S, b)$ has a tree decomposition $(T, \mathcal{X})$ of width at most 4 and size at most $s = 1 - 2m + 2\sum_{i=1}^{3m} w_i$, then there is a partition of the set $\{1, \ldots, 3m\}$ into m sets $S_1, \ldots, S_m$ such that $\sum_{i \in S_j} w_i = b$ for each $j = 1, \ldots, m$.*

Proof. Lemma 2.6 in [8] proved that if $G(S, b)$ has a path decomposition $(T, \mathcal{X})$ of width at most 4 and length at most $1 - 2m + 2\sum_{i=1}^{3m} w_i$, then there is a partition of the set $\{1, \ldots, 3m\}$ into m sets $S_1, \ldots, S_m$ such that $\sum_{i \in S_j} w_i = b$ for each $j = 1, \ldots, m$. So it is enough to prove that any tree decomposition (T, X) of $G(S, b)$ of width at most 4 and size at most $s = 1 - 2m + 2\sum_{i=1}^{3m} w_i$ is a path decomposition of $G(S, b)$.

As proved in Lemma 2.3 of [8], each bag in (T, X) contains exactly one of the cliques $K_3^{i,q}, K_4^{i,q}, K_5^j$. Indeed, each of these cliques has size at least 3. Moreover, any two of them share at most one vertex, and no two cliques of size 3 ($K_3^{i,q}$) share a vertex. So each bag of $(T, \mathcal{X})$ contains at most one of the cliques $K_3^{i,q}, K_4^{i,q}, K_5^j$. However, for any clique, there is a bag in $(T, \mathcal{X})$ containing its vertices. Since s equals the number of the cliques $K_3^{i,q}, K_4^{i,q}, K_5^j$, each bag of $(T, \mathcal{X})$ contains exactly one of them.

Moreover let us prove that any edge in $K_4^{i,q}, K_5^j, P_b^j$ (i.e. both the two endpoints of the edge) is contained in exactly one bag. Since each bag in (T, X) contains exactly one of the cliques $K_3^{i,q}, K_4^{i,q}, K_5^j$, the two endpoints of any edge in the paths $P_b^1, \ldots, P_b^m$ are contained in a bag containing some $K_3^{i,q}$. (The bags containing a $K_4^{i,q}$ (resp., K_5^j) cannot add another two vertices (resp., one vertex) since $(T, \mathcal{X})$ is a tree decomposition of width at most 4.) Every bag containing some $K_3^{i,q}$ contains at most one edge in the paths $P_b^1, \ldots, P_b^m$, because the bag can add at most another two vertices and any $K_3^{i,q}$ and P_b^j are disjoint. There are mb edges in the paths $P_b^1, \ldots, P_b^m$ and there are mb bags containing some $K_3^{i,q}$, so every bag containing a $K_3^{i,q}$ contains exactly one edge in the paths $P_b^1, \ldots, P_b^m$. So any edge in the paths $P_b^1, \ldots, P_b^m$ is contained in exactly one bag. Also each bag containing some $K_3^{i,q}$ contains 5 vertices, so it does not contains any edge (i.e. both its two endpoints) in $K_4^{i,q}$ or K_5^j. Therefore, any edge on $K_4^{i,q}, K_5^j$ is contained in exactly one bag.

Now we prove that there are only two leaves in T and so T is a path. If a bag containing some $K_3^{i,q}$ and an edge uv on some path P_b^j is a leaf bag in T, then its neighbor bag also contains u, v because both u and v are incident to



(a) F (b) $H_{2,4}$

Fig. 3. Example of the new gadget in $G(S, b)$.

other edges in $G(S, b)$. This is a contradiction with any edge (its two endpoints) on P^j_b are contained only in one bag. So any bag containing some $K^{i,q}_3$ is not a leaf bag in T. Similarly, we can prove that any bag containing any $K^{i,q}_4$ or K^j_5 for $1 < j < m+1$ is not a leaf bag in T. Thus there are only two bags containing K^1_5 and K^{m+1}_5 are leaves in T.

Then we get the following corollary.

Corollary 1. *It is NP-complete to compute s_4 in the class of graphs of treewidth at most* 4.

Theorem 1 follows from Lemma 1 and Corollary 1.
We furthermore modify the reduction to prove theorem 2.

Theorem 2. *It is NP-complete to compute s_4 in the class of planar graphs of treewidth at most* 3.

Proof. As in the previous reduction, we build a graph $G(S, b)$ for an instance of 3-PARTITION; we keep the subgraphs H_i as they are and modify the graph $H_{m,b}$ as follows. We replace the $m+1$ copies of K_5 by $m+1$ copies of the graph F that consists of a K_4 and a K_3 sharing an edge as depicted in Figure 3(a). We denote the copies by $F_1, F_2, \ldots, F_{m+1}$. The new graph $G(S, b)$ we obtain is planar and has treewidth 3.

Lemma 2 is still true and for 3 to be correct we need to prove that if $G(S, b)$ has a tree decomposition $(T, \mathcal{X})$ of width at most 4 and size at most $s = 1 - 2m + 2\sum_{i=1}^{3m} w_i$, then there is a bag of $(T, \mathcal{X})$ containing F_i, for each F_i, $i \in 1, \ldots, m+1$. Let us denote by K^i_3 and K^i_4 the two cliques sharing exactly one edge that form F_i. Each of these cliques, should appear in one bag. Note that among all the cliques of $G(S, b)$, the only cliques that can coexist in a bag are of the form K^i_3 and K^i_4 since the sum of the number of vertices of any other two cliques is more than 5. Let us suppose that there exists $j \in 1, \ldots, m+1$ such that no bag of $(T, \mathcal{X})$ contains F_j, i.e. K^j_3 and K^j_4 are not in the same bag. In this case the number of bags of $(T, \mathcal{X})$ is at least the number of the cliques $K^{i,q}_3, K^{i,q}_4, K^{i'}_4$ $(i' \neq j)$, plus the two bags containing K^j_3 and K^j_4. This gives a size of at least $2 - 2m + 2\sum_{i=1}^{3m} w_i$ wich is not possible.

3 Notations and preliminaries

In this section, we present the definitions and notations used throughout the report and some well-known facts about tree-decompositions.

3.1 Notations

Given a graph $G = (V, E)$, for any $S \subseteq V$, For an integer $c \geq 0$, a graph $G = (V, E)$ is *c-connected* if $|V| > c$ and no subset $V' \subseteq V$ with $|V'| < c$ is a separator in G. A *2-connected component* of G is a maximal 2-connected subgraph.

Let $(T, \mathcal{X})$ be any tree-decomposition of G. Abusing the notations, we will identify a node $t \in V(T)$ and its corresponding bag $X_t \in \mathcal{X}$. This means that, e.g., instead of saying $t \in V(T)$ is adjacent to $t' \in V(T)$ in T, we can also say that $X_t \in \mathcal{X}$ is adjacent to $X_{t'} \in \mathcal{X}$ in T. A bag $B \in \mathcal{X}$ is called a *leaf-bag* if B has degree one in T. Let $k \geq 1$ and G be a graph with $tw(G) \leq k$. A subset $B \subseteq V(G)$ is a *k-potential-leaf* if there is a tree-decomposition $(T, \mathcal{X})$ with width at most k and size $s_k(G)$ such that B is a leaf bag of $(T, \mathcal{X})$. A subgraph $H \subseteq V$ is a *k-potential-leaf* of G if $V(H)$ is a k-potential-leaf of G. Note that a k-potential-leaf has size at most $k+1$. Given a class of graphs $\mathcal{C}$ and integer $k \in \mathbb{N}^*$, a set of graphs $\mathcal{P}$ is called a *complete set of k-potential-leaves* of $\mathcal{C}$, if for any graph $G \in \mathcal{C}$, there exists a graph $H \in \mathcal{P}$ such that H is a k-potential-leaf of G.

A tree-decomposition is *reduced* if no bag is contained in another one. It is straightforward that, in any leaf-bag B of reduced tree-decomposition, there is $v \in V$ such that v appears only in B and so $N[v] \subseteq B$. Note that it implies that any reduced tree-decomposition has at most $n-1$ bags.

In the following we define two *transformation rules*, that take a tree-decomposition $(T, \mathcal{X})$ of a graph G, and computes another one without increasing the width nor the size.

Leaf. Let $X \in \mathcal{X}$ and $N_T(X) = \{X_1, \cdots, X_d\}$. Assume that, for any $1 < i \leq d$, $X_i \cap X \subseteq X_1$. Let $(T^*, \mathcal{X}^*) = Leaf(X, X_1, (T, \mathcal{X}))$ denote the tree-decomposition of G obtained by replacing each edge $X_iX \in E(T)$ by an edge X_iX_1 for any $1 < i \leq d$. Note that X becomes a leaf-bag after the operation. See in Fig. 4.

Reduce. Let $XX' \in E(T)$ with $X \subseteq X'$. Let $(T^*, \mathcal{X}^*) = Reduce(X, X', (T, \mathcal{X}))$ denote the tree-decomposition of G obtained by deleting the bag X from the tree-decomposition $Leaf(X, X', (T, \mathcal{X}))$. Note that the size of the tree decomposition is decreased by one after the operation.

From any tree-decomposition of G with width k and size s, it is easy to obtain a reduced tree-decomposition of G with width at most k and size at most $s-1$ by applying the Reduce operation while it is possible (i.e., while a bag is contained in another one). In particular, any minimum size tree decomposition is reduced.

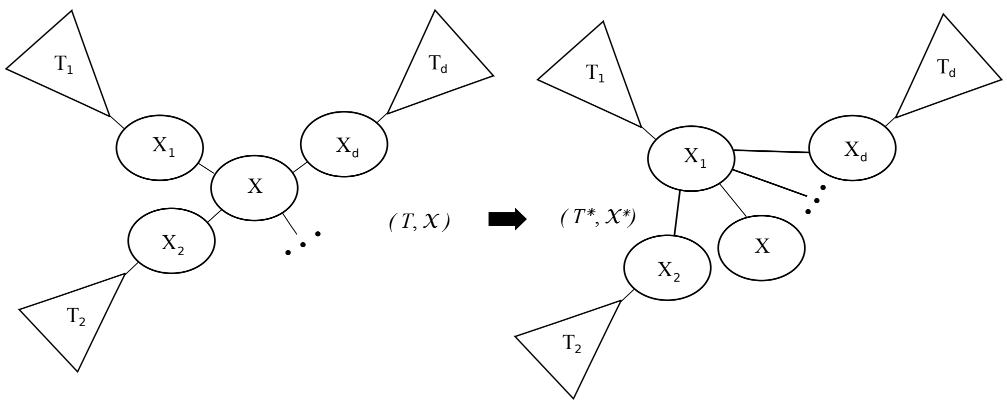


Fig. 4. In a tree decomposition $(T, \mathcal{X})$, $N_T(X) = \{X_1, \cdots, X_d\}$ and for any $1 < i \leq d$, $X_i \cap X \subseteq X_1$. For $1 \leq i \leq d$, $T_i \cup X_i$ induces the subtree containing X_i in $T \setminus \{XX_i\}$. Replace each edge $X_iX \in E(T)$ by an edge X_iX_1 for any $1 < i \leq d$. This gives a tree decomposition $(T^*, \mathcal{X}^*) = Leaf(X, X_1, (T, \mathcal{X}))$. X is a leaf-bag in $(T^*, \mathcal{X}^*)$.

We conclude this section by a general lemma on tree-decompositions. This lemma is known as folklore, we recall it for completness.

Lemma 4. *Let $(T, \mathcal{X})$ be a tree decomposition of a graph G. Let $X \in \mathcal{X}$ and $v, w \in X$. If there exists a connected component in $G \setminus X$ containing a neighbor of v and a neighbor of w, then there is a neighbor bag of X in $(T, \mathcal{X})$ containing v and w.*

Proof. First, let us note that, for any connected subgraph H of G, the set of bags of T that contain a vertex of H induces a subtree of T (the proof is easy by induction on $|V(H)|$).

Let C be a connected component in $G \setminus X$ containing a neighbor of v and a neighbor of w. By above remark, let T'_C be the subtree of T induced by the bags that contain some vertex of C. Moreover, because no vertices of C are contained in the bag X, then T'_C is a subtree of $T \setminus X$. Let T_C be the connected component of $T \setminus X$ that contains T'_C. Let $Y \in V(T_C)$ be the bag of T_C which is a neighbor of X in T. Let $x \in N(v) \cap C$ be a neighbor of v in C. Then there exists a bag $Z \in \mathcal{X}$ in T_C containing both x and v. So both X and Z contain vertex v. Then the bag Y, which is on the path between X and Z in T, also contains v. Similarly, we can prove that $w \in Y$.

Corollary 2. *Let $(T, \mathcal{X})$ be a tree decomposition of a 2-connected graph G. Let $X \in \mathcal{X}$ and $|X| \leq 2$. Then there is a neighbor bag Y of X in $(T, \mathcal{X})$ such that $X \subseteq Y$.*

Proof. Since G is 2-connected, $|V(G)| \geq 3$. So there exist at least another bag except X in $\mathcal{X}$.

If $|X| = 1$, let $X = \{v\}$. Then there is a neighbor bag Y of X containing v, since G is 2-connected and v is adjacent to some vertices in G. So $X \subseteq Y$.

Otherwise $X = 2$ and let $X = \{v, w\}$. Let G_1 be any connected component in $G \setminus X$. If v is not adjacent to any vertex in G_1, then $\{w\}$ separates $V(G_1)$ from $\{v\}$. It contradicts with the assumption that G is 2-connected. So any connected component in $G \setminus X$ containing a neighbor of v and a neighbor of w. From Lemma 4, there is a neighbor bag Y of X containing v, w, i.e. $X \subseteq Y$.

3.2 General approach

In what follows, we propose polynomial-time algorithms to compute minimum-size tree-decompositions of graphs with small treewidth. Our algorithms mainly use the notion of potential-leaf.

Let $k \geq 1$ and $G = (V, E)$ be a graph with $tw(G) \leq k$. The key idea of our algorithms is to identify a finite complete set of potential-leaves. Then, our algorithms are recursive: given a graph G and a k-potential-leaf H from the complete set, we compute a minimum-size tree-decomposition of G by adding H to a minimum-size tree-decomposition of a smaller graph.

The next lemmas formalize the above paragraph.

Lemma 5. *Let $k \geq 1$ and $G = (V, E)$ be a graph with $tw(G) \leq k$. Let $B \subseteq V$ be a k-potential-leaf of G. Let $S \subset B$ be the set of vertices of B that have a neighbor in $V \setminus B$. Then $s_k(G) = s_k(G_S \setminus (B \setminus S)) + 1$.*

Proof. Let us first prove $s_k(G) \leq s_k(G_S \setminus (B \setminus S)) + 1$. Suppose that $(T_S, \mathcal{X}_S)$ is a minimum size tree decomposition of width at most k of the graph $G_S \setminus (B \setminus S)$. Then there exists a bag $X \in \mathcal{X}_S$ containing S because S induces a clique in the graph $G_S \setminus (B \setminus S)$. So add the bag B and make it adjacent to X in the tree decomposition $(T_S, \mathcal{X}_S)$. Then we obtain a tree decomposition of width at most k for graph G of size $s_k(G_S \setminus (B \setminus S)) + 1$.

Now we prove that $s_k(G) \geq s_k(G_S \setminus (B \setminus S)) + 1$. Let $(T, \mathcal{X})$ be a minimum size tree decomposition of G of width at most k such that B is a leaf bag in it. Note that, if $B = V$ then $G_S \setminus (B \setminus S)$ is the empty graph. Let us assume that $B \subset V$. Then $(T, \mathcal{X})$ is also a tree decomposition of G_S. Let B be adjacent to the bag Y in $(T, \mathcal{X})$. Then $S \subset Y$ since each vertex in S is contained in another bag in $(T, \mathcal{X})$. Let $(T', \mathcal{X}')$ be the tree decomposition obtained by deleting the vertices in $B \setminus S$ in all the bags of $(T, \mathcal{X})$. Then B is changed to $B' = S \in \mathcal{X}'$ and let Y be changed to $Y' \in \mathcal{X}'$. So $B' \subseteq Y'$. Then the tree decomposition $Reduce(B', Y', (T', \mathcal{X}'))$ is a tree decomposition of $G_S \setminus (B \setminus S)$ of size $s_k(G) - 1$. So $s_k(G) - 1 \geq s_k(G_S \setminus (B \setminus S))$.

This lemma implies the following corollary:

Corollary 3. *Let $k \in \mathbb{N}^*$ and $\mathcal{C}$ be the class of graphs with treewidth at most k. If there is a $g(n)$-time algorithm $\mathcal{A}_k$ that, for any n-vertex-graph $G \in \mathcal{C}$, computes a k-potential-leaf of G. Then s_k can be computed in $O(g(n) \cdot n)$ time in the class of n-vertex graphs in $\mathcal{C}$. Moreover, a minimum size tree decomposition of width at most k can be constructed in the same time.*

Proof. Let $G \in \mathcal{C}$ be a n-vertex-graph. Let us apply Algorithm $\mathcal{A}_k$ to find a subgraph H of G in $g(n)$ time, which is a k-potential-leaf of G. Let $S \subset V(H)$ be the set of vertices having a neighbor in $G \setminus H$ and $G' = G_S \setminus (V(H) \setminus S)$. Then, by Lemma 5, $s_k(G) = s_k(G') + 1$. Finally, $|V(G')| \leq n - 1$ and G' has treewidth at most k. We then proceed recursively. So the total time complexity is $O(g(n) \cdot n)$. Moreover, for any minimum size $(s_k(G'))$ tree decomposition $(T', \mathcal{X}')$ of G' of width k, there is a bag X containing S since S induces a clique in G'. Add a new bag $N = V(H)$ adjacent to X in $(T', \mathcal{X}')$. The obtained tree decomposition is a minimum size ($s_k(G) = s_k(G') + 1$) tree decomposition of G of width at most k.

4 Graphs with treewidth at most 2

In this section, we describe algorithm $\mathcal{A}_2$ computes a 2-potential-leaf of a given graph. In particular, all graphs considered in this section have treewidth at most 2, i.e. partial 2-trees. Please see a complete set of 2-potential-leaf of graphs of treewidth at most 2 in Fig. 5. We are going to prove that any of the subgraphs in Fig. 5 is a potential-leaf and then that each non-empty graph of treewidth at most 2 contains one of them as a 2-potential-leaf.

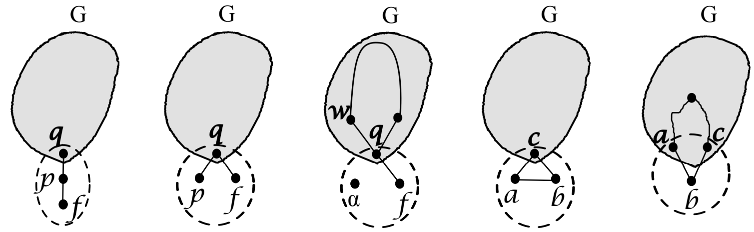


Fig. 5. Complete set of 2-potential-leaves of graphs of treewidth at most 2.

Lemma 6. *Let G be a graph with treewidth at most 2 and $p \in V(G)$ such that $N(p) = \{f, q\}$ and f has degree one (see in Fig. 5(a)). Then $\{f, p, q\}$ is a 2-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any tree-decomposition of G with width at most 2 and size at most $s \geq 1$. We show how to modify it to obtain a tree-decomposition with width at most 2 and size at most s and in which $\{f, p, q\}$ is a leaf bag.

Since $fp \in E(G)$, there is a bag B in $(T, \mathcal{X})$ containing both f and p. We may assume that B is the single bag containing f (otherwise, delete f from any other bag). Similarly, since $pq \in E(G)$, let X be a bag in $(T, \mathcal{X})$ containing both p and q.

First, let us assume that $X = B = \{f, p, q\}$. In that case, we may assume that X is the single bag containing p (otherwise, delete p from any other bag). If X is a leaf bag, then the lemma is proved. Otherwise, let $X_1, \cdots, X_d$ be the neighbors of X in T. Since f and p appear only in X, then $X \cap X_i \subseteq \{q\}$ for any $1 \leq i \leq d$. If there is $1 \leq i \leq d$ such that $q \in X_i$, let us assume w.l.o.g., that $q \in X_1$. By definition of the operation $Leaf$, the tree-decomposition $Leaf(X, X_1, (T, \mathcal{X}))$ has width at most 2, same size as $(T, \mathcal{X})$, and X is a leaf.

Second, consider the case when $X \neq B$. There are two cases to be considered. Either $B = \{f, p\}$ or $B = \{f, p, x\}$ with $x \neq q$. In the latter case, note that there is another bag B', neighbor of B, that contains x unless x is an isolated vertex of G. In the former case or if x appears only in B (in which case, x is an isolated vertex), let B' be any neighbor of B. Let $(T', \mathcal{X}')$ be obtained by deleting f, p in all bags of $(T, \mathcal{X})$. Then, contract the edge BB' in T', i.e., remove B and make any neighbor of B adjacent to B'. Note that, in the resulting tree-decomposition of $G \setminus \{f, p\}$, there is a bag X' containing q and with $|X'| \leq 2$ (the bag that results from X). Finally, add a bag $\{f, p, q\}$ adjacent to X' and, if node x was only in B, then add x to X'. The result is the desired tree-decomposition.

Lemma 7. *Let G be a graph with treewidth at most 2 and $q \in V(G)$ such that q has at least two one-degree neighbors f and p (see in Fig. 5(b)). Then $\{f, p, q\}$ is a 2-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any tree-decomposition of G with width at most 2 and size at most $s \geq 1$. We show how to modify it to obtain a tree-decomposition with width at most 2 and size at most s and in which $\{f, p, q\}$ is a leaf bag.

Since $fq \in E(G)$, there is a bag B in $(T, \mathcal{X})$ containing both f and q. We may assume that B is the single bag containing f (otherwise, delete f from any other bag). Similarly, since $pq \in E(G)$, let X be a bag in $(T, \mathcal{X})$ containing both p and q. Again, we may assume that X is the single bag containing p (otherwise, delete p from any other bag).

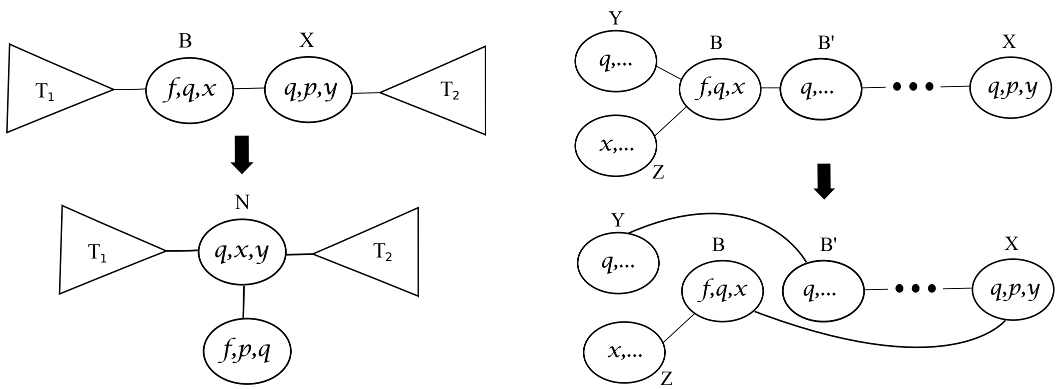
First, let us assume that $X = B = \{f, p, q\}$. If X is a leaf bag, then the lemma is proved. Otherwise, let $X_1, \cdots, X_d$ be the neighbors of X in T. Since f and p appear only in X, then $X \cap X_i \subseteq \{q\}$ for any $1 \leq i \leq d$. If there is $1 \leq i \leq d$ such that $q \in X_i$, let us assume w.l.o.g., that $q \in X_1$. By definition of the operation $Leaf$, the tree-decomposition $Leaf(X, X_1, (T, \mathcal{X}))$ has width at most 2, same size as $(T, \mathcal{X})$, and X is a leaf.

Second, let us assume that $X = \{f, q\}$ or $B = \{p, q\}$. In the former case, remove p from any bag and add p to X. In the latter case, remove f from any bag and add f to B. In both cases, we get a bag $\{f, p, q\}$ as in the first case.

Otherwise, let $B = \{f, q, x\}$, $x \neq p$, and $X = \{p, q, y\}$, $y \neq f$.

- If B and X are adjacent in T, then add a new bag $N = \{q, x, y\}$; remove B and X and make each of their neighbors adjacent to the new bag N and, finally, add a leaf-bag $\{f, p, q\}$ adjacent to N. See in Fig. 6(a). The obtained tree-decomposition has the desired properties.

– Otherwise, if there is a neighbor B' of B with $q, x \in B'$, then remove B, make all neighbors of B adjacent to B' and finally add a leaf-bag $\{f, p, q\}$ adjacent to X. The obtained tree-decomposition has the desired properties.
– Otherwise, let B' be the neighbor of B on the path between B and X. In this case, $q \in B'$ and $x \notin B'$. Moreover, q does not belong to any neighbor of B that contains x and the other way around: x does not belong to any neighbor of B that contains q. For any neighbor Y of B with $q \in Y$ (and hence $x \notin Y$), replace the edge $YB \in E(T)$ with the edge YB'. Finally, replace the edge $BB' \in E(T)$ by the edge BX. See in Fig. 6(b). In the resulting tree-decomposition of G, B and X are adjacent and we are back to the first item.



(a) In the tree decomposition $(T, \mathcal{X})$, let $T_1 \cup B$ (resp. $T_1 \cup B$) induce the subtree containing B (resp. X) in $T \setminus \{BX\}$. Delete B and X and make each of their neighbors adjacent to the new bag $N = \{q, x, y\}$; and add a leaf-bag $\{f, p, q\}$ adjacent to N.

(b) To be simple and clear, we show only the induced path from B to X in T and two neighbors $Y, Z \neq B'$ of B. Y contains q and Z contains x. Then we just change Y to be adjacent to B' instead of B and change B to be adjacent to X instead of B'.

Fig. 6. Explanation of proof of Lemma 7.

Lemma 8. *Let G be a graph with treewidth at most 2 and $q \in V(G)$ such that q has one neighbor f with degree 1 and for any vertex $w \in N(q) \setminus \{f\}$, $\{w, q\}$ belongs to a 2-connected component of G.*

If G has an isolated vertex α, then $\{q, f, \alpha\}$ is a 2-potential-leaf; otherwise $\{q, f\}$ is a 2-potential-leaf (see in Fig. 5(c)).

Proof. Let $(T, \mathcal{X})$ be any tree-decomposition of G with width at most 2 and size at most $s \geq 1$. We show how to modify it to obtain a tree-decomposition with width at most 2 and size at most s and in which $\{f, q, \alpha\}$ is a leaf bag if G has an isolated vertex α; and otherwise $\{f, q\}$ is a leaf bag.

Since $fq \in E(G)$, there is a bag B in $(T, \mathcal{X})$ containing both f and q. We may assume that B is the single bag containing f (otherwise, delete f from any other bag).

1. If $B = \{f, q\}$, then the intersection of B and any of its neighbor in T is empty or $\{q\}$. If there is a neighbor of B containing q, then let X be such a neighbor; otherwise let X be any neighbor of B. By definition of the operation $Leaf$, the tree-decomposition $Leaf(B, X, (T, \mathcal{X}))$ has width at most 2, same size as $(T, \mathcal{X})$, and B is a leaf. If there are no isolated vertices, we are done. Otherwise, if there is an isolated vertex α in G, then delete α in all bags of the tree-decomposition $Leaf(B, X, (T, \mathcal{X}))$ and add α to bag B, i.e. make $B = \{f, p, \alpha\}$. The result is the desired tree decomposition.
2. Otherwise let $B = \{f, q, x\}$.
   (a) If x is a neighbor of q, then x and q are in a 2-connected component of G. So there exists a connected component in $G \setminus B$ containing a vertex adjacent to x and a vertex adjacent to q. From Lemma 4, there is

a neighbor X of B in $(T, \mathcal{X})$ containing both x and q. Then by definition of the operation $Leaf$, the tree-decomposition $Leaf(B, X, (T, \mathcal{X}))$ has width at most 2, same size as $(T, \mathcal{X})$, and B is a leaf. Then delete x in B, i.e. $B = \{f, q\}$. Finally, if α is an isolated vertex of G, remove it to any other bag and add it to B. The result is the desired tree decomposition.

(b) Otherwise x is not adjacent to q. If there is a neighbor X of B in $(T, \mathcal{X})$ containing both x and q, then $(T, \mathcal{X})$ is modified as in case 2a. Otherwise, any neighbor of B in $(T, \mathcal{X})$ contains at most one of q and x.
If there is a neighbor of B in T containing q, then let Y be such a neighbor of B; otherwise let Y be any neighbor of B. Delete the edges between B and all its neighbors not containing x except Y in $(T, \mathcal{X})$ and make them adjacent to Y.
If there is no neighbor of B containing x, then x is an isolated vertex and we get a tree decomposition of the same size and width as $(T, \mathcal{X})$, in which there is a leaf bag $B = \{f, q, x\}$. It is a required tree decomposition.
Otherwise let Z be a neighbor of B in $(T, \mathcal{X})$ containing x, then delete the edges between B and all its neighbors containing x except Z in $(T, \mathcal{X})$ and make them adjacent to Z. Now B has only two neighbors Y and Z and $B \cap Y \subseteq \{q\}$, $B \cap Z = \{x\}$ and $Y \cap Z = \emptyset$. Delete the edge between B and Z and make Z adjacent to Y. Delete x in B, i.e. make $B = \{f, q\}$. See the transformations in Fig. 7. Then we get a tree decomposition of the same size and width as $(T, \mathcal{X})$, in which $B = \{f, q\}$ is a leaf bag. Again, if α is an isolated vertex of G, remove it to any other bag and add it to B. The result is the desired tree decomposition.

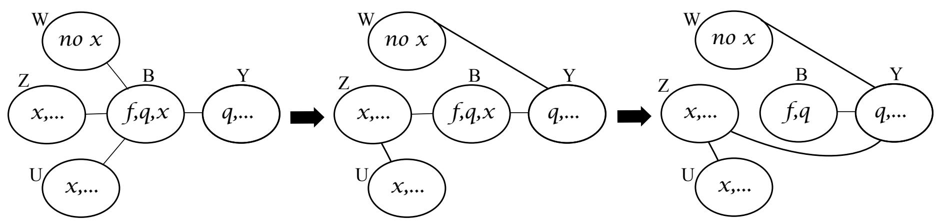


Fig. 7. To be simple and clear, we show only the subtree induced by B, Y and another three neighbors Z, W, U of B. Y contains q; Z, U both contain x and W does not contain x. First we make the bag not containing x, e.g. W adjacent to Y instead of B; and make the bag containing x except Z, e.g. U adjacent to Z instead of B. Second, make Z adjacent to Y instead of B and delete x in B. Then $B = \{f, q\}$ is a leaf-bag.

Lemma 9. *Let G be a graph of treewidth at most 2. Let $b \in V(G)$ with $N(b) = \{a, c\}$. If $N(a) = \{b, c\}$ (see in Fig. 5(d)) or if there is a path, with at least one internal vertex, between a and c in $G \setminus \{b\}$ (see in Fig. 5(e)), then $\{a, b, c\}$ is a 2-potential-leaf of G.*

Proof. Let $G = (V, E)$ be a graph of treewidth at most 2. Let $b \in V$ with exactly 2 neighbors $a, c \in V$ satisfy the hypotheses of the lemma. If $V = \{a, b, c\}$, the result holds trivially, so let us assume that $|V| \geq 4$.

Let $(T, \mathcal{X})$ be a reduced tree decomposition of width at most 2 of G. From $(T, \mathcal{X})$, we will compute a tree decomposition $(T^*, \mathcal{X}^*)$ of G without increasing the width or the size and such that $\{a, b, c\}$ is a leaf-bag of $(T^*, \mathcal{X}^*)$.

Let X be any bag of $(T, \mathcal{X})$ containing $\{a, b\}$ and Y be any bag containing $\{b, c\}$. The bags X, Y exist because $ab, bc \in E$. If $X = \{a, b\}$, then there exists a connected component in $G \setminus X$ containing a neighbor of a and a neighbor of b. By Lemma 4, there is a neighbor of X in $(T, \mathcal{X})$ that contains both a and b, contradicting the fact that $(T, \mathcal{X})$ is reduced. So $|X| = 3$ and, similarly, $|Y| = 3$.

- Let us first assume that $X = Y = \{a, b, c\}$. In particular, it is the case when $N(a) = \{b, c\}$ since $\{a, b, c\}$ induces a clique. We may assume that b only belongs to bag X (otherwise, remove b from any other bag).
  If $N(a) = \{b, c\}$, then we can also assume that a only belongs to X. Let Z be any neighbor of B containing c if exists; otherwise let Z be any neighbor of B (Z exists since $|V| \geq 4$). Otherwise, there exists a path P between a and c in $G \setminus \{b\}$ with at least one internal vertex. In this latter case, there exists a connected component in $G \setminus X$ containing a neighbor of a and a neighbor of c. So by Lemma 4, there is a neighbor bag Z of X in $(T, \mathcal{X})$ containing both a and c. In both cases, $Leaf(X, Z, (T, \mathcal{X}))$ is the desired tree-decomposition.

– Otherwise, $X = \{a, b, x\}$ and $Y = \{b, c, y\}$ with $x \neq c$ and $y \neq a$; and there exists a path P between a and c in $G \setminus \{b\}$ with at least one internal vertex. Let Q be the path between X and Y in $(T, \mathcal{X})$. We may assume that b only belongs to the bags in Q, because otherwise b can be removed from any other bag.
  - If X is adjacent to Y, then by properties of tree-decomposition, $X \cap Y$ separates a and c. Since $\{b\}$ does not separate a and c, $X \cap Y = \{b, x\}$, i.e. $x = y$. In this case, $(T^*, \mathcal{X}^*)$ is obtained by making $X = \{a, c, x\}$ and removing Y from $(T, \mathcal{X})$, then making all neighbors of Y adjacent to X and finally, adding a bag $\{a, b, c\}$ adjacent to X.
  - Otherwise, let X' be the bag in the path Q containing a, which is closest to Y. Similarly, let Y' be the bag in the path Q containing c, which is closest to X. Finally, let Q' be the path from X' to Y' in T and note that b belongs to each bag in Q' and a and c do not belong to any internal bag in Q'. Then we may assume that b only belongs to the bags in Q', because otherwise b can be removed from any other bag.
    If X' and Y' are adjacent in T, the proof is similar to the one in previous item. Otherwise, let Z be the neighbor of X' in Q'. By properties of tree-decompositions, $X' \cap Z$ separates a and c. Since $\{b\}$ does not separate a and c, let $X' \cap Z = \{b, x'\}$. Since $Z \neq \{b, x'\}$ because $(T, \mathcal{X})$ is reduced, then $Z = \{b, x', z\}$ for some $z \in V$. Replace b with a in all the bags. By doing this $(T, \mathcal{X})$ is changed to a tree decomposition $(T^c, \mathcal{X}^c)$ of the graph G/ab obtained by contracting the edge ab in G. In $(T^c, \mathcal{X}^c)$, the bag X' has become $X^c = \{a, x'\}$ and Z is changed to be $Z^c = \{a, x', z\}$. So X^c can be reduced in $(T^c, \mathcal{X}^c)$. Moreover Y is changed to $Y^c = \{a, c, y\}$. To conclude, let us add the bag $\{a, b, c\}$ adjacent to Y^c in the tree decomposition $Reduce(X^c, Z^c, (T^c, \mathcal{X}^c))$. See in Fig. 8. The result is the desired tree-decomposition $(T^*, \mathcal{X}^*)$ of G.

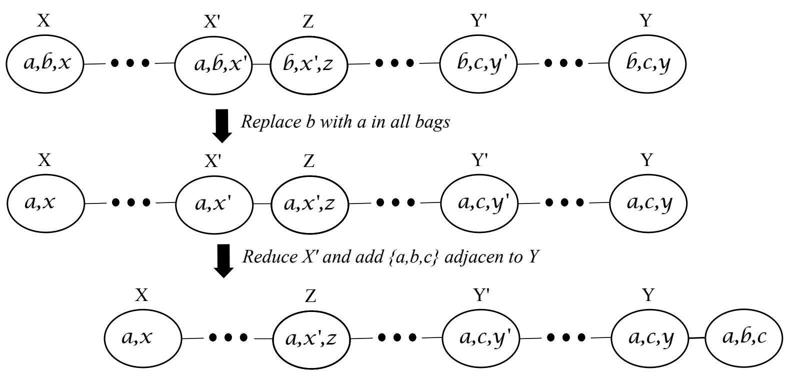


Fig. 8. To be simple and clear, we show only the path from X to Y. After the two transformations, $\{a, b, c\}$ is a leaf-bag.

Before going further, let us introduce some notations. A *bridge* in a graph $G = (V, E)$ is any subgraph induced by two adjacent vertices u and v of G (i.e., $uv \in E$) such that the number of connected components strictly increases when deleting the edge uv, but not the two vertices u, v in G, i.e., $G' = (V, E \setminus \{uv\})$ has strictly more connected components than G. A vertex $v \in V$ is a *cut vertex* if $\{v\}$ is a separator in G. A maximal connected subgraph without a cut vertex is called a *block*. Thus, every block of a graph $G = (V, E)$ is either a 2-connected component of G or a bridge or an isolated vertex. Conversely, every such subgraph is a block. Different blocks of G intersect in at most one vertex, which is a cut vertex of G. Hence, every edge of G lies in a unique block, and G is the union of its blocks.

Let $G = (V, E)$ be a connected graph and let $r \in V$. A spanning tree T of G is a BFS-tree of G if for any $v \in V(G)$, the distance from r to v in G is the same as the one in T. Let $\mathcal{B} = \{C : C \text{ is a block of } G\}$. The *block graph* of G is the graph $B(G)$ whose vertices are the blocks of G and two block-vertices of $B(G)$ are adjacent if the corresponding blocks intersect, that is, $B(G) = (\mathcal{B}, \{C_1C_2 : C_1, C_2 \in \mathcal{B} \text{ and } C_1 \cap C_2 \neq \emptyset\})$. Note that $B(G)$ is connected. Finally, a *block-tree* of G is any BFS-tree F (with any arbitrary root) of $B(G)$. See an example in Fig. 9.

There is a linear (in the number of edges) algorithm for computing all blocks in a given graph [10]. Also a BFS-tree can be found in linear (in the number of vertices plus the number of edges) time. So given a graph $G = (V, E)$, we can compute a block tree F of G in $O(|V| + |E|)$ time.

Now we are ready to prove next theorem by using the Lemmas 6-9.

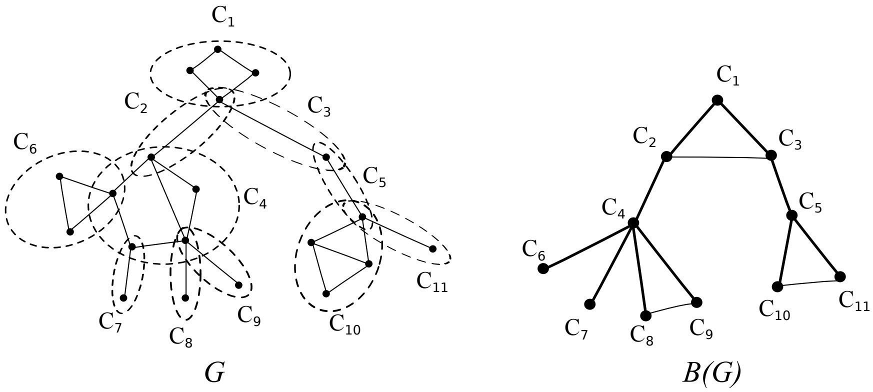


Fig. 9. Graph G is connected. For $i = 1, \ldots, 11$, each C_i is a block of G. $B(G)$ is the block graph of G. The BFS tree of $B(G)$ with bold edges is a block tree of G with root C_1.

Theorem 3. *There is an algorithm that, for any n-vertex-m-edge-graph G with treewidth at most 2, computes a 2-potential-leaf of G in time $O(n + m)$.*

Proof. If $n \leq 3$, then $V(G)$ is a 2-potential-leaf of G. Let us assume that $n \geq 4$. First, let us compute the set of isolated vertices in G, which can be done in $O(n)$ time. If G has only isolated vertices, then any three vertices induce a 2-potential-leaf of G. Otherwise, there is at least one edge in G.

Let G_1 be any connected component of G containing at least one edge. If $|V(G_1)| = 2$, then from Lemma 9, either G has an isolated vertex α and $\{\alpha, u, v\}$ is a 2-potential-leaf or $\{u, v\}$ is a 2-potential leaf.

Otherwise, $|V(G_1)| \geq 3$. Compute a block tree F of G_1 rooted in an arbitrary block R. This can be done in time $O(n + m)$. Note that any node in F corresponds to either a 2-connected component of G or a bridge $uv \in E(G)$. Let C be a leaf block in F, which is furthest from R and $|V(C)|$ is maximum. There are several cases to be considered.

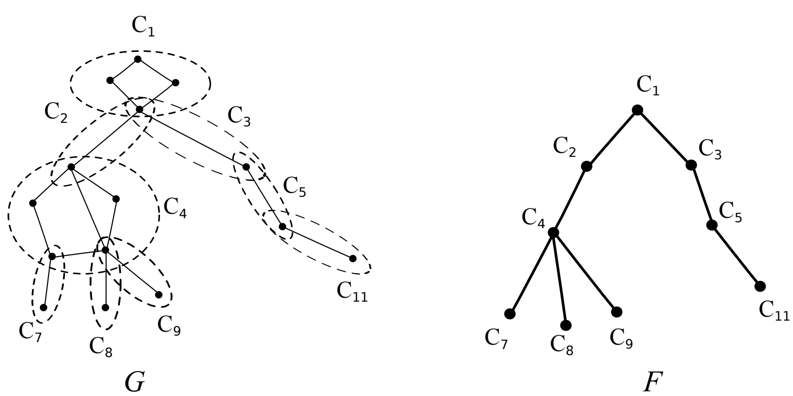


Fig. 10. This graph G is an induced subgraph of the graph in Fig. 9. Its block tree F, with root C_1, has two blocks less than the one in Fig. 9 (the blocks C_6 and C_{10}). All leaf blocks, C_7, C_8, C_9, C_{11}, in F contains two vertices of G.

– let us first assume that C is a bridge in G, i.e. C consists of one edge $fp \in E(G)$ and p is a cut vertex. Then f has degree one in G because C is a leaf block in F. Let P be the parent block of C in F. Then any child block A of P in F consists of one edge because C has the maximum number of vertices among all the children of P; and A is a leaf block in F because C is a furthest leaf from the root block R.

If P has another child block except C in F containing the cut vertex p, then this child block also consists of one edge $f'p \in E(G)$, where f' has degree one in G because this child is also a leaf block in F. (For example, in Fig. 10, take C as C_8, which intersects C_9 with a cut vertex.) From Lemma 7, $\{f, p, f'\}$ is a 2-potential-leaf.
Otherwise P has only one child block C in F containing the cut vertex p. Then any vertex in $N_G(p) \setminus \{f\}$ belongs to P. If P is also a bridge in G, i.e., P consists of one edge $pq \in E(G)$, then p has degree 2 in G. (For example, in Fig. 10, take C as C_{11}, whose parent C_5 is also a bridge in G.) From Lemma 6, $\{f, p, q\}$ is a 2-potential-leaf of G. Otherwise, P is a 2-connected component of G and $p \in V(G)$ satisfies the hypothesis of Lemma 8. (For example, in Fig. 10, take C as C_7, whose parent C_4 is a 2-connected component of G.) Hence, either G has an isolated vertex α and $\{\alpha, f, p\}$ is a 2-potential-leaf or $\{f, p\}$ is a 2-potential-leaf.
– Finally, let us assume that C is a 2-connected component of G. It is known that any graph with at least two vertices of treewidth k contains at least two vertices of degrees at most k [5]. There is no degree one vertex in C because C is 2-connected. So there exists two vertices with degree 2 in C. Since C is a leaf in F, there is only one cut vertex of G in C. So there exists a vertex b in C which has degree two in G. If $|V(C)| \geq 4$, then there exists a path between two neighbors a, c of b in $G \setminus \{b\}$ containing at least one internal vertex. (For example, in Fig. 9, take C as C_{10}.) From Lemma 9, $\{a, b, c\}$ is a 2-potential-leaf. Otherwise C is a triangle $\{a, b, c\}$ with at least two vertices with degree 2 in G. Again from Lemma 9, $\{a, b, c\}$ is a 2-potential-leaf.

So the total time complexity is $O(n + m)$.

Corollary 4. *s_2 can be computed in polynomial-time in general graphs. Moreover, a minimum size tree decomposition can be constructed in polynomial-time in the class of partial 2-trees.*

Proof. Let G be any graph. It can be checked in polynomial-time whether $tw(G) \leq 2$ (e.g. see [17]). If $tw(G) > 2$, then $s_2 = \infty$. Otherwise $tw(G) \leq 2$, then the result follows from Theorem 3 and Corollary 3.

5 Minimum-size tree-decompositions of width at most 3

In this section, we study the computation of s_3 in the class of trees and 2-connected outerplanar graphs.

5.1 computation of s_3 in trees

In this subsection, given a tree G, we show how to find a 3-potential-leaf in G. We characterize a complete set of 3-potential-leaves of trees in Fig. 11. We first prove that each of the subgraphs in Fig. 11 is a 3-potential-leaf and then that any tree with at least four vertices contains one of them.

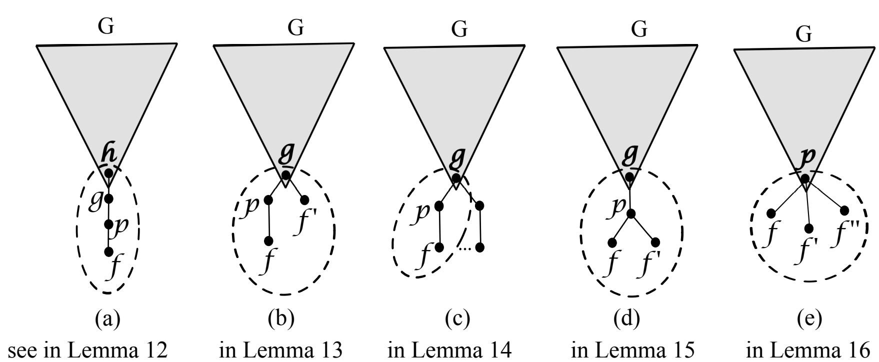


Fig. 11. Complete set of 3-potential-leaves of trees.

Lemma 10. *Let $(T, \mathcal{X})$ be a tree decomposition of a tree G. Let $X \in \mathcal{X}$ and $N_T(X) = \{X_1, \ldots, X_d\}$ for $d \geq 1$. Suppose that for any $1 \leq i \leq d$, $X_i \cap X \subseteq \{x\}$. Then there is a tree decomposition $(T', \mathcal{X}')$ of G of the same width and size as $(T, \mathcal{X})$ such that X is a leaf bag.*

Proof. If there is a bag X_i for $1 \leq i \leq d$ containing x, then let B be X_i. Otherwise let B be any neighbor of X. By definition of the operation $Leaf$, the tree-decomposition $Leaf(X, B, (T, \mathcal{X}))$ is a desired tree decomposition.

Lemma 11. *Let G be a tree rooted at $r \in V(G)$. Let f be a leaf in G. Let p be the parent of f and let g be the parent of p in G. Let p have degree 2 in G. Let $(T, \mathcal{X})$ be a tree decomposition of G of width at most 3 and size at most $s \geq 1$. If there is no bag in $(T, \mathcal{X})$ containing all of f, p, g, then there is a tree decomposition $(T', \mathcal{X}')$ of G of width at most 3 and size at most s such that $\{f, p, g\} \in \mathcal{X}'$ is a leaf bag.*

Proof. Since $fp \in E(G)$, there is a bag B in $(T, \mathcal{X})$ containing both f and p. We may assume that B is the single bag containing f (otherwise, delete f from any other bag). Similarly, since $pg \in E(G)$, let X be a bag in $(T, \mathcal{X})$ containing both p and g. Let P be the path in T from B to X. Then p is contained in all bags on P and we may assume that p is not contained in any other bags (otherwise, delete p from any other bag). Let B' be the neighbor of B on P. Then $B \cap B' \supseteq \{p\}$. Note that it is possible that $B' = X$.

If $B = \{f, p\}$, then make all other neighbors of B adjacent to B' and delete B. Add a bag $\{f, p, g\}$ adjacent to X. The result is a desired tree decomposition $(T', \mathcal{X}')$.

Otherwise, B contains at least one vertex not in $\{f, p\}$. If $B \cap B' = \{p\}$, then $\{p\}$ separates g from any vertex in $B \setminus \{p\}$. So $B \setminus \{p\} = \{f\}$, i.e., $B = \{f, p\}$. It contradicts with the assumption.

So $|B \cap B'| \geq 2$ and let $\{p, x\} \subseteq B \cap B'$. Then create a bag $Z = (B \setminus \{f, p\}) \cup (B' \setminus \{p, x\})$. (Note that $x \in Z$ since $x \in B$.) So $|Z| \leq 4$. Make Z adjacent to all neighbors of B and all neighbors of B'; and delete the two bags B and B'; and delete f, p from all bags. Finally add another new bag $N = \{f, p, g\}$ adjacent to some bag containing g. The obtained tree decomposition has width at most 3, same size as $(T, \mathcal{X})$, and a bag $N = \{f, p, g\}$.

Lemma 12. *Let G be a tree rooted at $r \in V(G)$ and $|V(G)| \geq 4$. Let f be a leaf in G. Let p be the parent of f and let g be the parent of p in G. Suppose that both p and g have degree 2. Let h be the parent of g (see in Fig. 11(a)), then $H = G[\{f, p, g, h\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any reduced tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{f, p, g, h\}$ is a leaf bag.

From Lemma 11, we can assume that there is a bag B in $(T, \mathcal{X})$ containing all f, p, g. We may assume that B is the single bag containing f, p (otherwise, delete f, p from any other bag). Since $gh \in E(G)$, let Y be a bag in $(T, \mathcal{X})$ containing both h and g.

1. If $B = Y = \{f, p, g, h\}$, then the intersection of B and any of its neighbor in T is contained in $\{h\}$. A desired tree decomposition can be obtained from Lemma 10.
2. If $B = \{f, p, g\}$, then the intersection of B and any of its neighbors in T is contained in $\{g\}$. From Lemma 10, there is a tree-decomposition $(T', \mathcal{X}'))$ of the same width and size as the ones of $(T, \mathcal{X})$ such that $B = \{f, p, g\}$ is a leaf. Then delete B in the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$ and add a new bag $N = \{f, p, g, h\}$ adjacent to Y. The obtained tree decomposition has the desired properties.
3. Otherwise, $B = \{f, p, g, x\}$ where $x \neq h$. Then the intersection of B and any of its neighbor in T is contained in $\{g, x\}$. Let P be the path in T from B to Y. Then g is contained in all bags on P. Let B' be the neighbor of B on P. Note that it is possible that $B' = Y$. If $B \cap B' = \{g\}$, then $\{g\}$ separates h from x. So $x \in \{f, p\}$ i.e. $B = \{f, p, g\}$, a contradiction with the assumption. So we have $B \cap B' = \{g, x\}$. By definition of the operation $Leaf$, the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, p, g, x\}$ is a leaf. Then delete B in the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$ and add a new bag $N = \{f, p, g, h\}$ adjacent to Y. The obtained tree decomposition has the desired properties since $\{g, x\} \subseteq B'$ and $\{g, h\} \subseteq Y$.

Lemma 13. *Let G be a tree rooted at $r \in V(G)$ and $|V(G)| \geq 4$. Let f a leaf in G. Let p be the parent of f and let g be the parent of p in G. If p has degree 2 and g has a child f', which is a leaf in G (see in Fig. 11(b)), then $H = G[\{f, p, g, f'\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any reduced tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{f, p, g, f'\}$ is a leaf bag.

From Lemma 11, we can assume that there is a bag B in $(T, \mathcal{X})$ containing all f, p, g. We may assume that B is the single bag containing f, p (otherwise, delete f, p from any other bag). Since $gf' \in E(G)$, let Y be a bag in $(T, \mathcal{X})$ containing both f and g. We may assume that Y is the single bag containing f' (otherwise, delete f' from any other bag).

- If $B = Y = \{f, p, g, f'\}$, then the intersection of B and any of its neighbor in T is contained in $\{g\}$. A desired tree decomposition can be obtained from Lemma 10.
- If $B = \{f, p, g\}$, then delete f' in Y and add f' in B, then we are in the previous case.
- Otherwise, $B = \{f, p, g, x\}$ where $x \neq f'$. Then the intersection of B and any of its neighbor in T is contained in $\{g, x\}$. Let P be the path in T from B to Y. Then g is contained in all bags on P. Let B' be the neighbor of B on P. If $B \cap B' = \{g, x\}$, then by definition of the operation $Leaf$, the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, p, g, x\}$ is a leaf. Then in the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$, delete f' in Y and remove x from B and add f' to B, i.e. make $B = \{f, p, g, f'\}$. The obtained tree decomposition has the desired properties since $\{g, x\} \subseteq B'$.
  Otherwise $B \cap B' = \{g\}$. Delete f' from the bag Y and add x in Y; delete x from B and add f' in B, i.e., make $B = \{f, p, g, f'\}$; finally make all neighbors of B except B' adjacent to Y since now $\{g, x\} \subseteq Y$. The result is the desired tree decomposition.

Lemma 14. *Let G be a tree rooted at $r \in V(G)$ and $|V(G)| \geq 3$. Let f be one of the furthest leaves from r. Let p be the parent of f and let g be the parent of p in G. If g has degree at least 3 and any child of g has degree 2 in G (see in Fig. 11(c)), then $H = G[\{f, p, g\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any reduced tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{f, p, g, f'\}$ is a leaf bag.

From Lemma 11, we can assume that there is a bag B in $(T, \mathcal{X})$ containing all f, p, g. We may assume that B is the single bag containing f, p (otherwise, delete f, p from any other bag).

1. If $B = \{f, p, g\}$, then the intersection of B and any of its neighbor in T is contained in $\{g\}$. A desired tree decomposition can be obtained from Lemma 10.
2. Otherwise, $B = \{f, p, g, x\}$. Then the intersection of B and any of its neighbor in T is contained in $\{g, x\}$.
   (a) If there is a neighbor B' of B such that $B \cap B' = \{g, x\}$, then by definition of the operation $Leaf$, the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, p, g, x\}$ is a leaf. Then delete x in B in the tree-decomposition $Leaf(B, B', (T, \mathcal{X}))$ since $\{g, x\} \subseteq B'$. The obtained tree decomposition has the desired properties.
   (b) Otherwise any neighbor of B contains at most one of g and x. If x is not adjacent to g, then there is a connected component in $G \setminus B$ containing a neighbor of g and a neighbor of x. From Lemma 4, there exists a neighbor bag of B in $(T, \mathcal{X})$ containing g and x. It is a contradiction. So we have x is adjacent to g in this case.
      i. x is a child of g. Then x has exactly one child y, which is a leaf in G since f is one of the furthest leaves from r. Since $yx \in E(G)$, there is a bag Y in $(T, \mathcal{X})$ containing both y and x. We may assume that Y is the single bag containing y (otherwise, delete y from any other bag). Since $\{g, x\} \subset B$ and any neighbor of B contains at most one of g and x, any bag except B contains at most one of g and x. Then $g \notin Y$ because $x \in Y$. So y, x, g are not contained in one bag. From Lemma 11, we can modify $(T, \mathcal{X})$ to obtain a tree-decomposition $(T', \mathcal{X}')$ of width at most 3 and size at most s having a leaf bag $X = \{y, x, g\}$. Note that x (resp. y) plays the same role as p (resp. f) in G, i.e., g, p, f and g, x, y are symmetric in G. Hence, the result is a desired tree decomposition.
      ii. x is the parent of g. Let p' be another child of g and let f' be the child of p', which is a leaf in G. Let B' be the bag in $(T, \mathcal{X})$ containing both f' and p'. We may assume that B' is the single bag containing f' (otherwise, delete f' from any other bag). Let X' be a bag containing both p' and g. Then we have

$X' \neq B$ (because $p' \notin B$). Since $g \in X'$ any bag except B contains at most one of g and x, we have $x \notin X'$. In the following, we modify $(T, \mathcal{X})$ to obtain a tree-decomposition $(T', \mathcal{X}')$ with width at most 3 and size at most s having a bag $\{f', p', g\}$. Then we are in case 1, since g, p, f and g, p', f' are symmetric in G.
If $B' = X' = \{f', p', g\}$ then, we are done. If $B' = X' = \{f', p', g, x'\}$. Then x' is not x, which is the parent of g, since $x \notin X'$. So we can do as in case 2a or case 2(b)i.
Otherwise, $B' \neq X'$. From Lemma 11, we can modify $(T, \mathcal{X})$ to obtain a tree-decomposition with width at most 3 and size at most s having a leaf bag $\{f', p', g\}$.

Lemma 15. *Let G be a tree rooted at $r \in V(G)$ and $|V(G)| \geq 4$. Let f a leaf in G. Let p be the parent of f. If p has exactly two children f, f' in G and let g be the parent of p in G (see in Fig. 11(d)), then $H = G[\{f, f', p, g\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any reduced tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{f, f', p, g\}$ is a leaf bag.

Since $fp \in E(G)$, there is a bag B in $(T, \mathcal{X})$ containing both f and p. We may assume that B is the single bag containing f (otherwise, delete f from any other bag). Similarly, let B' be the single bag in $(T, \mathcal{X})$ containing both f' and p. Let X be a bag containing both p and g.

1. If $B = B' = X = \{f, f', p, g\}$, then we can assume that B is the single bag containing p (otherwise, delete p from any other bag). So the intersection of B and any of its neighbor in T is contained in $\{g\}$. Then a desired tree decomposition can be obtained from Lemma 10.
2. If $B = B' = \{f, f', p\}$, then the intersection of B and any of its neighbor in T is contained in $\{p\}$. Let Y be a neighbor of B in T containing p. By definition of the operation $Leaf$, the tree-decomposition $Leaf(B, Y, (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, f', p\}$ is a leaf. Then delete B and add a new bag $N = \{f, f', p, g\}$ adjacent to X. The result is a desired tree decomposition.
3. If $B = B' = \{f, f', p, x\}$ and $x \neq g$, then the intersection of B and any of its neighbor in T is contained in $\{p, x\}$. Since $x \notin \{f, f', g\}$, p is not adjacent to x. There is a connected component in $G \setminus B$ containing a neighbor of p and a neighbor of x. From Lemma 4, there exists a neighbor bag of B in $(T, \mathcal{X})$ containing p and x. Let Y be such a neighbor of B in T. By definition of the operation $Leaf$, the tree-decomposition $Leaf(B, Y, (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, f', p, x\}$ is a leaf. Then delete x from B and we get a tree decomposition having a bag $\{f, f', p\}$. So we are in case 2.
4. If $B \neq B'$ and $|B| \leq 3$, then delete f' in B and add f' in B. Then we are in case 2 or 3. It is proved similarly if $B \neq B'$ and $|B'| \leq 3$.
5. Otherwise $B \neq B'$ and $|B| = |B'| = 4$. Let $B = \{f, p, x, y\}$ and $B' = \{f', p, x', y'\}$. Let P be the path in T from B to B'. Then p is contained in all bags on P. Let Y be the neighbor of B on P. If $B \cap Y = \{p\}$, then $\{p\}$ separates x from x'. But p is not a separator between any two vertices in $V(G) \setminus \{f, f'\}$. It is a contradiction. So w.l.o.g. we can assume that $B \cap Y \supseteq \{p, x\}$. Deleting f, f', p in all bags of $(T, \mathcal{X})$. Add a new bag $Z = \{x, y\} \cup Y \setminus \{p, x\}$ adjacent to all neighbors of the two bags B, Y and delete B and Y. Finally add another new bag $N = \{f, f', p, g\}$ adjacent to a bag containing g. The obtained tree decomposition has the desired properties.

Lemma 16. *Let G be a tree rooted at $r \in V(G)$ and $|V(G)| \geq 4$. Let all children of p be leaves in G and p have at least three children f, f', f'' (see in Fig. 11(e)). Then $H = G[\{p, f, f', f''\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any reduced tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{p, f, f', f''\}$ is a leaf bag.

Since $fp \in E(G)$, there is a bag B in $(T, \mathcal{X})$ containing both f and p. We may assume that B is the single bag containing f (otherwise, delete f from any other bag). Similarly, let B' (resp. B'') be the single bag in $(T, \mathcal{X})$ containing both f' (resp. f') and p.

1. If $B = B' = B'' = \{f, f', f'', p\}$, then the intersection of B and any of its neighbor in T is contained in $\{p\}$. A desired tree decomposition can be obtained from Lemma 10.

2. If $B = B' = \{f, f', p\}$, then delete f'' in B'' and add f'' in B. Then we are in case 1. It can be proved similarly if $B = B'' = \{f, f'', p\}$ or $B' = B'' = \{f', f'', p\}$.
3. If $B = B' = \{f, f', p, x\}$ and $x \neq f''$, then the intersection of B and any of its neighbor in T is contained in $\{p, x\}$.
   If x is a child of p, then x is also a leaf in G and x play the same role as f''. Then we are in case 1. So in the following we assume that x is not a child of p.
   If x is not the parent of p, then p is not adjacent to x. So there is a connected component in $G \setminus B$ containing a neighbor of p and a neighbor of x. From Lemma 4, there exists a neighbor bag of B in $(T, \mathcal{X})$ containing p and x. Let Y be such a neighbor of B in T. By definition of the operation $Leaf$, the tree-decomposition $Leaf(B, Y, (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, f', p, x\}$ is a leaf. Then delete x from B and we get a tree decomposition having a bag $\{f, f', p\}$. So we are in case 2.
   Otherwise x is the parent of p. Let P be the path in T from B to B''. Then p is contained in all bags on P. Let Y be the neighbor of B on P. If $B \cap Y = \{p, x\}$, then by definition of the operation $Leaf$, the tree-decomposition $Leaf(B, Y, (T, \mathcal{X}))$ has width at most 3, same size as $(T, \mathcal{X})$, and $B = \{f, f', p, x\}$ is a leaf. Then deleting x from B we are in case 2. Otherwise, $B \cap Y = \{p\}$. So $\{p\}$ separates x from all vertices in $B'' \setminus \{p\}$. Then all vertices in $B'' \setminus \{p\}$ are children of p and so they are leaves in G. So we can assume that any vertex in $B'' \setminus \{p\}$ are contained only in B'' (otherwise we can delete it in any other bags). Then delete f, f' from B and add vertices of $B'' \setminus \{f'', p\}$ in B; and make $B'' = \{f, f', f'', p\}$. Then we are in case 1.
   The cases $B = B'' = \{f, f'', p, x\}$ and $x \neq f'$ or $B' = B'' = \{f', f'', p, x\}$ and $x \neq f$ can be proved similarly.
4. Otherwise, none two of f, f', f'' are contained in a same bag.
   If $|B| \leq 3$, then delete f' in B' and add f' in B. Then we are in case 2 or 3. It is proved similarly if and $|B'| \leq 3$ or $|B''| \leq 3$.
   Otherwise $|B| = |B'| = |B''| = 4$. In the following, we are going to modify $(T, \mathcal{X})$ to obtain a tree-decomposition with width at most 3 and size at most s having a bag X containing at least two of f, f', f'' or $f \in X$ and $|X| \leq 3$. Then we are in the above cases. Note that all children of p play the same role (they are all leaves) in G. So it is enough to get that X contains at least two children of p or that X contains one child of p and $|X| \leq 3$.
   Let T_p be the subtree in T induced by all the bags containing p. If $|V(T_p)| \leq 2$, there exists one bag containing at least two children of p since p has at least three children. Then it is done. So we assume that $|V(T_p)| \geq 3$. There is a bag $R \in V(T_p)$ containing both p and g. Root T_p at R and let $L \in V(T_p)$ be one of the furthest leaf bag in T_p from R. If there is no child of p in L, then we can delete p in L and consider $T_p \setminus \{L\}$. So we can assume there is a vertex $l \in L$, which is a child of p in G. Let Y be the neighbor of L in T_p. If the intersection of $L \cap Y = \{p\}$, then p separate any vertex in $L \setminus \{p\}$ and any vertex in $Y \setminus \{p\}$. So at least one of L, Y, denoted as X, contains only p and children of p. Then either X contains at least two children of p or X contains only one children and $|X| = 2$. So $(T, \mathcal{X})$ and X satisfy the desired properties.
   Otherwise, $|L \cap Y| \geq 2$. If Y has no other child except L in T_p, then $Y \neq R$ since $|V(T_p)| \geq 3$. Let $X = \{p, l\}$ if Y contains no child of p; and $X = \{p, l, l'\}$ if Y contains one child l' of p. Add a new bag $Z = Y \cup L \setminus X$. Since $|Y \cap L| \geq 2$, $|Y \cup L| \leq 6$. Then $|Z| \leq 4$, since $X \subseteq Y \cup L$ and $|X| \geq 2$. Make Z adjacent to all neighbors of Y, L in T and delete Y, L. Finally make X adjacent to R. The obtained tree decomposition and X have the desired properties.
   Otherwise, Y has at least another child L' in T_p. Then L' is also a furthest leaf from R in T_p, since L is a furthest leaf from R. For the same reason as L, there is a vertex $l' \in L$, which is a child of p in G. Let $L = \{l, p, x, y\}$ and $L' = \{l', p, x', y'\}$. So the intersection of L (resp. L') and any of its neighbors except Y in T is contained in $\{x, y\}$ (resp. $\{x', y'\}$). Create a new bag $N = \{x, y, x', y'\}$ adjacent to all neighbor of L, L' and delete L, L'. Finally add another bag $X = \{p, l, l'\}$ adjacent to Y. The obtained tree decomposition and X have the desired properties.

From Lemmas 12- 16 and Corollary 3, we obtain the following result.

Corollary 5. *s_3 and a minimum size tree decomposition of width at most 3 can be computed in polynomial-time in the class of trees.*

Proof. From Corollary 3, it is enough to prove we can find a 3-potential-leaf in any tree in polynomial time.

Let G be any tree. If $|V(G)| \leq 4$, then $V(G)$ is a 3-potential-leaf. Let us assume that $|V(G)| \geq 5$. Root G at any vertex r. Let f be one of the furthest leaves from r in G. Let p, g, h be the first three vertices on the path from f to r in G (if exist), i.e. p is f's parent; g is p's parent; and h is g's parent in G.

- If g, p both have only one child in G, then $\{f, p, g, h\}$ is a 3-potential-leaf of G from Lemma 12;
- If p has only one child and g has a child f', which is a leaf in G, then $\{f, p, g, f'\}$ is a 3-potential-leaf of G from Lemma 13;
- If p has only one child and any child of g has exactly one child, then $\{f, p, g\}$ is a 3-potential-leaf of G from Lemma 14;
- If p has only one child and there exist a child p' of g, which has exactly two children f_1, f_2, then $\{f_1, f_2, p', g\}$ is a 3-potential-leaf of G from Lemma 15;
- If p has only one child and there exist a child p' of g, which has at least three children f_1, f_2, f_3, then $\{f_1, f_2, f_3, p'\}$ is a 3-potential-leaf of G from Lemma 16;
- If p has exactly two children f, f', then $\{f, f', p, g\}$ is a 3-potential-leaf of G from Lemma 15;
- Otherwise p has at least three children f, f', f'', then $\{f, f', f'', p\}$ is a 3-potential-leaf of G from Lemma 16.

In fact, the algorithm for trees can be extended to forests by consider their connected component, i.e., trees. The only difference is in Lemma 14 the 3-potential-leaf becomes $\{f, p, g, \alpha\}$ if there is an isolated vertex α in the given forest.

5.2 Computation of s_3 in 2-connected outerplanar graphs

In this subsection, given a 2-connected outerplanar graph G, we show how to find a 3-potential-leaf in G. See a complete set of 3-potential-leaves of 2-connected outerplanar graphs in Fig. 12. We first prove that each subgraph in the Fig. 12 is a 3-potential-leaf and then we show that any 2-connected outerplanar graphs contains one of them.

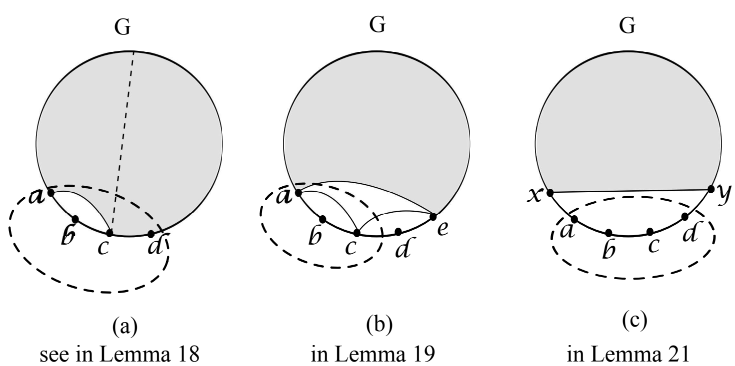


Fig. 12. Complete set of 3-potential-leaves of 2-connected outerplanar graphs.

The following fact is well known for 2-connected outerplanar graphs.

Lemma 17. *[15] A 2-connected outerplanar graph has the unique Hamilton cycle.*

In the rest of this subsection, let G be a 2-connected outerplanar graph and C be the Hamilton cycle in G.

Definition 2. *Any edge in $E(G) \setminus E(C)$ is called a chord in G.*

The vertices $v_1, \ldots, v_j \in V(G)$, for $2 \leq j \leq |V(G)|$, are *consecutive* in C (we also say that they are consecutive in G) if $v_i v_{i+1} \in E(C)$ for $1 \leq i \leq j-1$; and $v_1, \ldots, v_j \in V(G)$ are also called the consecutive vertices from v_1 to v_j.

Lemma 18. *Let $a, b, c, d \in V(G)$ be consecutive in C. If $\{a, b, c\}$ induces a clique and c has degree 3 in G (see in Fig. 12(a)), then $H = G[\{a, b, c, d\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{a, b, c, d\}$ is a leaf bag.

Since $\{a, b, c\}$ induces a clique in G, there is a bag B containing all a, b, c. Let X be a bag in $(T, \mathcal{X})$ containing both c and d (it exists since $cd \in E(G)$). Note that b is not incident to any chords, i.e. has degree 2. (Because if $by \in E(G)$ is a chord in G, then deleting all chords except ac, by in G and contracting the edges in C except ab, bc we get a K_4-minor in G. It is a contradiction with the fact that G is outerplanar.)

Replace b, c with a in all bags of $(T, \mathcal{X})$. Then $(T, \mathcal{X})$ becomes a tree decomposition $(T', \mathcal{X}')$ of the graph G' obtained by contracting the edges ab and bc. The bag X becomes X', which contains both a and d; and B becomes $B' = \{a\}$ if $B = \{a, b, c\}$ or $B' = \{a, x\}$ if $B = \{a, b, c, x\}$. From Corollary 2, in both case there exists a neighbor Y of B' such that $B' \subseteq Y$. So B' can be reduced in $(T', \mathcal{X}')$. The tree decomposition $Reduce(B', Y, (T', \mathcal{X}'))$ has one bag less than $(T, \mathcal{X})$. Finally, add a new bag $N = \{a, b, c, d\}$ adjacent to X', which contained both a and d, in the tree decomposition $Reduce(B', Y, (T', \mathcal{X}'))$. The result is a desired tree decomposition, because b, c are not adjacent to any vertices in $V(G) \setminus N$.

Lemma 19. *Let $a, b, c, d, e \in V(G)$ be consecutive in C. If $\{a, b, c\}$ and $\{c, d, e\}$ induce two cliques respectively in G and $ae \in E(G)$ (see in Fig. 12(b)), then $H = G[\{a, b, c\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify it to obtain a tree-decomposition with width at most 3 and size at most s and in which $\{a, b, c\}$ is a leaf bag.

Since $\{a, b, c\}$ (resp. $\{c, d, e\}$) induces a clique in G, there is a bag X (resp. Y) containing all a, b, c (resp. c, d, e). Note that b, c, d are not adjacent to any vertices in $V(G) \setminus \{a, b, c, d, e\}$.

Delete b, c, d in all bags of $(T, \mathcal{X})$. Then $(T, \mathcal{X})$ becomes a tree decomposition $(T', \mathcal{X}')$ of the graph $G' = G \setminus \{b, c, d\}$. The bag X becomes becomes $X' = \{a\}$ if $X = \{a, b, c\}$ or $X' = \{a, x\}$ if $B = \{a, b, c, x\}$. From Corollary 2, in both case there exists a neighbor A of X' such that $X' \subseteq A$. So X' can be reduced in $(T', \mathcal{X}')$. Similarly, the bag Y becomes Y', which can also be reduced in $(T', \mathcal{X}')$. After reducing the two bags X', Y' in $(T', \mathcal{X}')$, let the obtained tree decomposition be $(T'', \mathcal{X}'')$. Finally, add two new bags $N_1 = \{a, b, c\}$ and $N_2 = \{a, c, d, e\}$; make N_1 adjacent to N_2 and make N_2 adjacent to a bag Z containing both a and e in the tree decomposition $(T'', \mathcal{X}'')$. (Z exists because $ae \in E(G')$.) The result is a desired tree decomposition.

Lemma 20. *Let C_l be a cycle of $l \geq 4$ vertices. Let $(T, \mathcal{X})$ be a tree decomposition of C_l of width at most 3. Then there exist either a bag containing all vertices of $V(C_l)$ (only if $l = 4$) or two bags $X, Y \in \mathcal{X}$ such that X (resp. Y) contains at least three consecutive vertices x_1, x_2, x_3 (resp. y_1, y_2, y_3) and the two edge sets $\{x_1x_2, x_2x_3\} \cap \{y_1y_2, y_2y_3\} = \emptyset$.*

Proof. The treewidth of any cycle is bigger than 1, so there exists a bag in any tree decomposition of a cycle (with at least 4 vertices) containing two vertices not consecutive, equivalently they are not adjacent in the cycle. We prove the lemma by induction on l in the following.

First let us prove that it is true for $l = 4$. Let a, b, c, d be the four consecutive vertices in C_4. Let $(T, \mathcal{X})$ be a tree decomposition of width at most 3. Then there exists a bag containing a, c or b, d. W.l.o.g assume a, c are contained in one bag. So $(T, \mathcal{X})$ is also a tree decomposition of H, obtained from C_4 by adding the edge ac. The set $\{a, b, c\}$ induces a clique in H. So there is a bag X containing a, b, c. For the same reason, there is a bag Y containing c, d, a. If $X = Y$ then there is a bag containing all a, b, c, d of $V(C_4)$. Otherwise there are two bags X, Y such that $X \supseteq \{a, b, c\}$ and $Y \supseteq \{c, d, a\}$. We see that $\{ab, bc\} \cap \{cd, da\} = \emptyset$. So the lemma is true for $l = 4$.

Now suppose it is true for $l \leq n - 1$ and we prove it for $l = n \geq 5$. Note that since $(T, \mathcal{X})$ has width 3 and $l \geq 5$, there is no bag containing all vertices of $V(C_l)$. So in the following we prove that there always exist two bags X, Y with the desired properties. Let $v_1, \ldots, v_n$ be the n consecutive vertices in C_n. Let (T, X) be a tree-decomposition of width at most 3 of C_n. Then there exists a bag containing two non-adjacent vertices v_i, v_j for $1 \leq i < j \leq n$. So $(T, \mathcal{X})$ is also a tree decomposition of the graph H, obtained from C_n by adding the edge v_iv_j. The graph H is also the union of two subcycles C^1 induced by $\{v_i, \ldots, v_j\}$ and C^2 induced by $\{v_j, \ldots, v_n, \ldots, v_i\}$. Then $\max\{|C^1|, |C^2|\} \leq n-1$. Let (T^1, X^1) (resp. (T^2, X^2)) be the tree decomposition of C^1 (resp. C^2) obtained by deleting all vertices not in C^1 (resp. C^2) in the bags of (T, X).

If $|V(C^1)| = 3$ then there is a bag in (T^1, X^1) containing $V(C^1) = \{v_i, v_{i+1}, v_j = v_{i+2}\}$. So $v_iv_j \notin \{v_iv_{i+1}, v_{i+1}v_j\}$.

If $|V(C^1)| \geq 4$ then, by induction, there exist either a bag in (T^1, X^1) containing all vertices of $V(C^1) = \{v_i, v_{i+1}, v_{i+2}, v_j = v_{i+3}\}$ or two bags A, B in (T^1, X^1) containing three consecutive vertices a_1, a_2, a_3 and b_1, b_2, b_3

respectively in C^1; moreover, $\{a_1a_2, a_2a_3\} \cap \{b_1b_2, b_2b_3\} = \emptyset$. So we have either $v_iv_j \notin \{a_1a_2, a_2a_3\}$ or $v_iv_j \notin \{b_1b_2, b_2b_3\}$.

In both cases ($|V(C^1)| = 3$ and $|V(C^1)| \geq 4$), there is at least one bag X in (T^1, X^1) containing three consecutive vertices in C^1, denoted as x_1, x_2, x_3, such that $v_iv_j \notin \{x_1x_2, x_2x_3\}$. So x_1, x_2, x_3 are also consecutive in C. Similarly, there is at least one bag Y in (T^2, X^2) containing three consecutive vertices in C^2, denoted as y_1, y_2, y_3, such that $v_iv_j \notin \{y_1y_2, y_2y_3\}$. So y_1, y_2, y_3 are also consecutive in C. Finally, we have $\{x_1x_2, x_2x_3\} \cap \{y_1y_2, y_2y_3\} = \emptyset$ because $E(C^1) \cap E(C^2) = \{v_iv_j\}$ and $v_iv_j \notin \{x_1x_2, x_2x_3\}$.

Lemma 21. *Let xy be a chord in G. Let C' be the set of all the consecutive vertices from x to y in C and $|C'| \geq 4$. If each vertex in $C' \setminus \{x, y\}$ has degree 2 in G, then for any consecutive vertices $a, b, c, d \in C'$ (see in Fig. 12(c)), $H = G[\{a, b, c, d\}]$ is a 3-potential-leaf of G.*

Proof. Let $(T, \mathcal{X})$ be any tree decomposition of width at most 3 and size at most $s \geq 1$ of G. We show how to modify $(T, \mathcal{X})$ to obtain a tree-decomposition of G, which has width at most 3, size at most s and a leaf bag $\{a, b, c, d\}$.

Note that the vertices of C' induce a cycle in G. Without confusion, we denote this cycle C'. Let (T', X') be the tree decomposition of C' obtained by deleting all vertices not in C' in the bags of (T, X). From Lemma 20, there is either a bag containing all vertices in C' (only if $|C'| = 4$); or two bags X, Y containing three consecutive vertices in C' respectively and the two corresponding edge sets do not intersect.

In the former case, $V(C') = \{a, b, c, d\}$ and so $(T, \mathcal{X})$ is also a tree decomposition of $G \cup \{ac\}$, from Lemma 18, $\{a, b, c, d\}$ is a 3-potential-leaf of G.

In the latter case, let $X \supseteq \{u, v, w\}$ and $Y \supseteq \{u', v', w'\}$, where u, v, w (resp. u', v', w') are consecutive in C'. Since $\{uv, vw\} \cap \{u'v', v'w'\} = \emptyset$, we have either $xy \notin \{uv, vw\}$ or $xy \notin \{u'v', v'w'\}$. W.l.o.g. assume that $xy \notin \{uv, vw\}$. Then u, v, w are also consecutive in C and at least one of u, w has degree 2 in G. W.l.o.g. suppose w has degree 2 in G, i.e.$w \notin \{x, y\}$ (since x, y have degree at least 3 in G). Let $z \in C'$ be the other neighbor (except v) of w in C'. (z exists because $w \notin \{x, y\}$.)

$(T, \mathcal{X})$ is also a tree decomposition of $G \cup \{uw\}$, which is still an outerplanar graph by assumptions. Note that w has degree 3 in the graph $G \cup \{uw\}$. So from Lemma 18, we can modify $(T, \mathcal{X})$ to obtain a tree-decomposition $(T', \mathcal{X}')$ of $G \cup \{uw\}$, which has width at most 3, size at most s and a leaf bag L containing four consecutive vertices $\{u, v, w, z\}$. Note that $(T', \mathcal{X}')$ is also a tree decomposition of G. So we get a tree decomposition where a leaf bag contains 4 consecutive vertices of C'. It remains to show how to modify it to obtain a tree decomposition with a leaf bag $\{a, b, c, d\}$.

Let B be the neighbor of L in T. Then $u, z \in B$ since each of u, z is adjacent to some vertices in $G \setminus L$. We can assume that L is the single bag containing v, w in $(T', \mathcal{X}')$, because otherwise we can delete them in any other bags. Thus, deleting the bag L in $(T', \mathcal{X}')$, we get a tree decomposition $(T_1, \mathcal{X}_1)$ of the graph G_1, which is the graph obtained by deleting v, w and adding an edge uz in G. So $(T_1, \mathcal{X}_1)$ has width at most 3 and size at most $s-1$. Note that the graph G_1 is isomorphic to the graph $G_2 \equiv G \cup \{ad\} \setminus \{b, c\}$ since $z \in C'$. So from the tree decomposition $(T_1, \mathcal{X}_1)$ of G_1 we can obtain a tree decomposition $(T_2, \mathcal{X}_2)$ of G_2 with the same width and size. Note that since $ad \in E(G_2)$, there is a bag Y containing both a and d. Finally, add a new bag $N = \{a, b, c, d\}$ adjacent to Y in $(T_2, \mathcal{X}_2)$. The result is a desired tree decomposition.

Lemma 22. *There is an algorithm that, for any 2-connected outerplanar graph G, computes a* 3*-potential-leaf of G in polynomial time.*

Proof. Let G be a 2-connected outerplanar graph and C be the unique Hamilton cycle of G. If $|V(G)| \leq 4$, then $V(G)$ is a 3-potential-leaf of G. Otherwise, $|V(G)| \geq 5$ and consider the outerplanar embedding of G.

- If there exists an inner face f with at most one chord of G and f has at least four vertices, then from Lemma 21, the set of any four consecutive vertices in f, which are also consecutive in C, is a 3-potential-leaf in G.
- If there is an inner face $f = \{a, b, c\}$ with only one chord ac of G and c has degree 3, then let d be the other neighbor of c except b, a. From Lemma 18, the set of four consecutive vertices a, b, c, d, is a 3-potential-leaf in G.
- Otherwise, let $\mathcal{F}$ be the set of all inner faces with only one chord of G. Then any face $f \in \mathcal{F}$ has three vertices and both the two endpoints of the chord in f have degree at least 4, i.e., they are incident to some other chords except this one. We can prove by induction on $|V(G)|$ that:

Claim. There exist two faces $f_1, f_2 \in \mathcal{F}$ such that (1)$f_1 = \{a, b, c\}$; (2) $f_2 = \{c, d, e\}$; (3) a, b, c, d, e are consecutive in G; (4) there is a face f_0 containing both ac and ce and at most one chord, which is not in any face of $\mathcal{F}$. See in Fig. 13.

It is true when $|V(G)| = 5$. Assume that it is true for $|V(G)| \leq n-1$. Now we prove it is true for $|V(G)| = n$. Note that $\mathcal{F} \neq \emptyset$ if there is at least one chords in G, which is valid in this case. Let $f \in \mathcal{F}$ have three consecutive vertices x, y, z and let $xz \in E(G)$ be the single chord in f. Then the graph $G \setminus y$ is a 2-connected outerplanar graph with $n-1$ vertices. From the assumption, we have the desired faces f_0', f_1', f_2' in $G \setminus y$. If xz is not an edge in any face of f_1', f_2', then these faces are also the desired faces in G. Otherwise, let xz be an edge of f_1' or $f_2' = \{x, z, t\}$. Then z has degree 3 in G, i.e. it is not incident to any other chords except xz, since $xt \in E(G)$. So we are in second case above, which contradicts with the assumption.

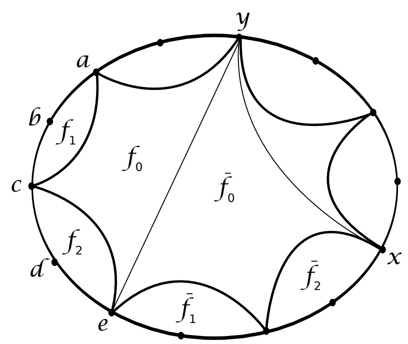


Fig. 13. $\mathcal{F}$ is the set of all inner faces with only one chord of G, such as $f_1, f_2, \bar{f}_1, \bar{f}_2$. The faces f_0, f_1, f_2 satisfy the properties in Claim 5.2. But $\bar{f}_0, \bar{f}_1, \bar{f}_2$ does not satisfy the properties since $\bar{f}_0$ contains two edges ey, xy which are not in any face of $\mathcal{F}$.

In the following, let f_0, f_1, f_2 be the faces as in Claim 5.2. If $ae \in E(G)$, then from Lemma 19, $\{a, b, c\}$ is a 3-potential-leaf of G.
Otherwise, we can prove that any tree decomposition of G of width at most 3 can be modified to a tree decomposition of $G \cup \{ae\}$ with the same width and size in the following. So $\{a, b, c\}$ is a 3-potential-leaf of G.
Let $(T, \mathcal{X})$ be a tree decomposition of width at most 3 and size at most $s \geq 1$ of G. Let $(T_0, \mathcal{X}_0)$ be the tree decomposition obtained by deleting all vertices not in f_0. Then $(T_0, \mathcal{X}_0)$ is a tree decomposition of f_0 (without confusion f_0 is used to denote the face and the cycle induced by vertices in f_0 as well). From Lemma 20, there is a bag containing three consecutive vertices u, v, w in f_0 and uv, vw are edges of some faces in $\mathcal{F}$. (Note that u, v, w are not consecutive in C.) So $(T, \mathcal{X})$ is also a tree decomposition of $G \cup uw$. The graph $G \cup uw$ and the graph $G \cup ae$ are isomorphic. So from $(T, \mathcal{X})$ we can obtain a tree decomposition $(T', \mathcal{X}')$ of $G \cup ae$ with the same width and size. Then $(T', \mathcal{X}')$ is the desired tree decomposition.

From Lemmas 22 and Corollary 3, we obtain the following result.

Corollary 6. *s_3 can be computed and a minimum size tree decomposition of width at most 3 can be constructed in polynomial-time in the class of 2-connected outerplanar graphs.*

6 Conclusion

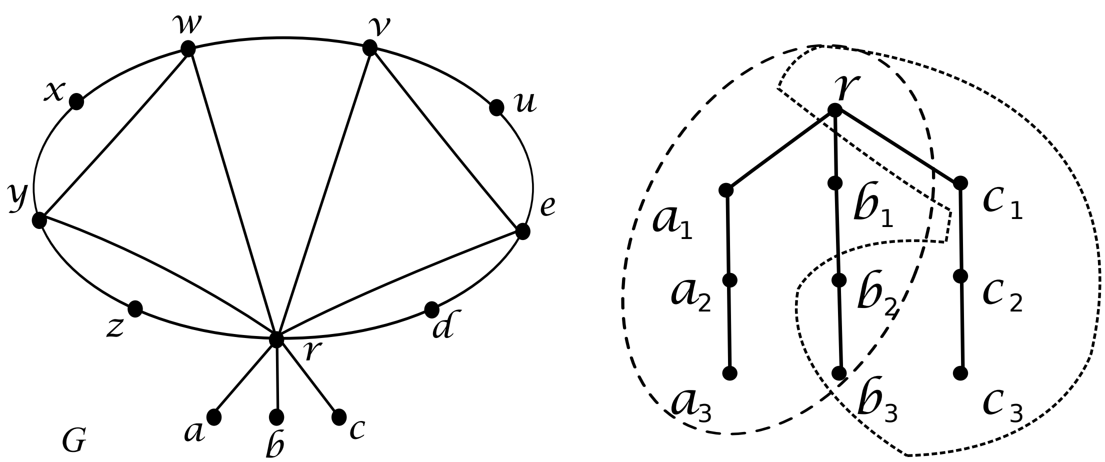
In this report, we gave preliminary results on the complexity of minimizing the size of tree-decompositions with given width. Table 1 summarizes our results as well as the remaining open questions.

We currently investigate the case of s_3 in the class of connected graphs with treewidth 2 or 3 and we conjecture it is polynomially solvable. But it is more tricky than computing s_3 in trees and 2-connected outerplanar graphs. It seems that a global view of the graph needs to be considered to decide wether a subgraph is a 3-potential-leaf of the graph.

	s_1	s_2	s_3	s_4	$s_k, k = \max\{tw+1, 5\}$
Graphs of treewidth at most $tw = 1$	P (trivial)	P	P	?	?
Graphs of treewidth at most $tw = 2$	-	P	?	?	?
Graphs of treewidth at most $tw = 3$	-	-	?	NP-hard	?
Graphs of treewidth at most $tw \geq 4$	-	-	-	NP-hard	NP-hard

Table 1. Summary of the complexity results.

See an example in Fig. 14(a). In this example, G is a connected outerplanar graph. $\{r, a, b, c\}$ is not a 3-potential-leaf of G, but it is a 3-potential-leaf of $G \setminus \{yw\}$. Let $G' \equiv G \setminus \{a, b, c\}$. Then G' is 2-connected outerplanar. From the algorithm of computing s_3 in 2-connected outerplanar graphs in subsection 5.2, we can compute that $s_3(G') = 5$. So if $\{r, a, b, c\}$ is a potential-leaf of G, then $s_3(G) = 6$. But there exists a tree decomposition of G of width 3 and size 5, where the bags are $\{a, r, z, y\}, \{r, y, x, w\}, \{b, r, w, v\}, \{r, v, u, e\}, \{c, r, d, e\}$. So $\{r, a, b, c\}$ is not a 3-potential-leaf of G. However, in the graph $G'' \equiv G \setminus \{yw\}$, we can prove that $s_3(G'') = 5$ and there is a minimum size tree decomposition containing $\{r, a, b, c\}$ as a leaf bag, i.e. $\{r, a, b, c\}$ is 3-potential-leaf of G''. So the existence of the edge yw, not incident to any vertex in $\{r, a, b, c\}$, changes the behavior of $\{r, a, b, c\}$.



(a) $\{r, a, b, c\}$ is not a 3-potential-leaf of G, but it is a 3-potential-leaf of $G \setminus \{yw\}$. The five bags $\{a, r, z, y\}, \{r, y, x, w\}, \{b, r, w, v\}, \{r, v, u, e\}, \{c, r, d, e\}$ connecting as a path in this order forms a tree decomposition of G.

(b) In any minimum size tree decomposition of width 5 (and size 2) of this tree, there exists a bag inducing a forest, not a subtree. For example, in a tree decomposition of width 5 and size 2, one bag is $\{r, a_1, a_2, a_3, b_1, b_2\}$ and the other one is $\{r, b_2, b_3, c_1, c_2, c_3\}$.

Fig. 14.

The problem of computing s_k, for $k \geq 4$, seems more intricate already in the case of trees. Indeed, our polynomial-time algorithms to compute s_k, $k \leq 3$, in trees mainly rely on the fact that, for any tree T, there exists a minimum-size tree-decomposition of T with width at most 3, where each bag induces a connected subtree. This is unfortunately not true anymore in the case of tree-decomposition with width 5. As an example, consider the tree G (with 10 nodes) obtained from a star with three 3 leaves by subdividing twice each edge. See in Fig. 14(b). $s_5(G) = 2$ and any minimum size tree decomposition has a bag X such that $G[X]$ is disconnected.

References

1. ARNBORG, S., CORNEIL, D. G., AND PROSKUROWSKI, A. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic*

Discrete Methods 8, 2 (Apr. 1987), 277–284.
2. ARNBORG, S., AND PROSKUROWSKI, A. Characterization and recognition of partial 3-trees. *SIAM J. Algebraic Discrete Methods 7*, 2 (Apr. 1986), 305–314.
3. BODLAENDER, H. L. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput. 25*, 6 (1996), 1305–1317.
4. BODLAENDER, H. L., DRANGE, P. G., DREGI, M. S., FOMIN, F. V., LOKSHTANOV, D., AND PILIPCZUK, M. A $o(c^k n)$ 5-approximation algorithm for treewidth. *CoRR abs/1304.6321* (2013).
5. BODLAENDER, H. L., AND KOSTER, A. M. C. A. Treewidth computations ii. lower bounds. *Inf. Comput. 209*, 7 (2011), 1103–1119.
6. COURCELLE, B. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation 85*, 1 (1990), 12 – 75.
7. DEMAINE, E. D., AND HAJIAGHAYI, M. The bidimensionality theory and its algorithmic applications. *Comput. J. 51*, 3 (2008), 292–302.
8. DERENIOWSKI, D., KUBIAK, W., AND ZWOLS, Y. Minimum length path decompositions. *CoRR abs/1302.2788* (2013).
9. GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
10. HOPCROFT, J., AND TARJAN, R. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM 16*, 6 (June 1973), 372–378.
11. KAJITANI, Y., ISHIZUKA, A., AND UENO, S. Characterization of partial 3-trees in terms of three structures. *Graphs and Combinatorics 2*, 1 (1986), 233–246.
12. MATOUSEK, J., AND THOMAS, R. Algorithms finding tree-decompositions of graphs. *Journal of Algorithms 12*, 1 (1991), 1 – 22.
13. ROBERTSON, N., AND SEYMOUR, P. D. Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms 7*, 3 (1986), 309–322.
14. SANDERS, D. P. On linear recognition of tree-width at most four. *SIAM J. Discret. Math. 9*, 1 (1996), 101–117.
15. SYSTO, M. M. Characterizations of outerplanar graphs. *Discrete Mathematics 26*, 1 (1979), 47 – 53.
16. WALD, J. A., AND COLBOURN, C. J. Steiner trees, partial 2-trees, and minimum ifi networks. *Networks 13*, 2 (1983), 159–167.
17. WALD, J. A., AND COLBOURN, C. J. Steiner trees, partial 2-trees, and minimum ifi networks. *Networks 13*, 2 (1983), 159–167.